



Universidad de las Ciencias Informáticas
Facultad 8

**Sistema de Control para la
Gestión Legal de Software Educativo**



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

Autores:

Julio Cesar García Mosquera

Yunesti Pérez La Rosa

Tutor:

Ing. Delmis Caridad Báez Deniz

Ciudad de la Habana

Julio del 2007

“Año 49 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los __ días del mes de _____ del año _____.

Julio Cesar García Mosquera

Yunesti Pérez La Rosa

Firma del autor

Firma del autor

Delmis Caridad Báez Deniz

Firma del tutor

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.”

Albert Einstein. Físico y Matemático alemán.

AGRADECIMIENTOS

A nuestra Tutora Delmis por enseñarnos a ser informáticos más capaces.

A nuestro hermanito Sandy, por ayudarnos con el diseño.

A Pedro Luís Basulto, por ayudarnos en todo lo que nos hizo falta y por su amistad.

A nuestros padres, simplemente, por ser nuestros padres.

A Amaray e Isabel por estar siempre a nuestro lado y por su infinito cariño.

A Yudita por ayudarnos con la Ingeniería de Software.

A Abduly, Dayron, Osiris, y demás compañeros de aula.

A Alexander y Julio por llevarnos a las canchas a despejar cuando lo necesitamos.

A Febe Ángel Ciudad por ayudarnos con sus conocimientos.

A los amigos de Colón por ayudarnos con la ortografía.

A la revolución y a nuestro Comandante en Jefe que nos ha dado esta maravillosa oportunidad.

A todos, Muchas Gracias.

DEDICATORIA

Yo, Yunesti Pérez La Rosa, les dedico este trabajo a mis padres que siempre han estado a mi lado y que tanto me han enseñado en esta vida, a los cuales debo lo que soy. A mi Hermanita linda a la que tanto quiero. A tía Tata y a tío Robe, a mis abuelos, a mi familia. A mis amigos más jóvenes como Carlitin, Addrey, Sandy, Julio, Alexander, Alerseyv, Leyvis, y a todos los que de una forma u otra han demostrado su amistad. A los más viejitos, Norma, Jose, La China, Fagundo, Julián y todos los demás que se puedan quedar. Muy en especial a Nidia, mi profesora de la secundaria. A mis profesores de Universidad, Tony y Basulto. En fin, a todas las personas que de alguna forma han ocupado un lugar especial en mi vida. A mi novia Isabel por estar siempre a mi lado y quererme tanto.

Yo, Julio, dedico este trabajo especialmente a mis padres Francisco y Bertha que tanto han dado en esta vida por mi, a mis hermanas queridas Mildrey y Nelly por ayudarme en todo lo que han podido, a mi tía Alga y a mi tío Norberto por brindarme su ayuda y cariño desde que empecé, a mi abuela Estrella, a mi abuelo Francisco y a mi abuela Antonia que aunque ya no estén, me quisieron mucho y siempre estarán presente, a mi tío Alexis, a mi tía carmen, a mi tía Landy, a mi primo Guillermito que siempre me ayudado, en general a toda mi familia, a mis amigos de infancia que siempre me acompañaron y ayudaron Randolf, Yoandy, Yandy, Dayan, Yusniel a mis amigos de la universidad, Yunesti, Sandy, a mi amigo Basulto(le debo mi primer salario) que mas no pudo hacer por mi, a todos mis amigos en general, que son muchos. A mi novia Amaray por apoyarme, quererme y darme todo su amor todo este tiempo y no porque esté de última quiere decir que ocupe ese lugar sino que quise guardar un espacio especial para ella.

RESUMEN

El presente trabajo tiene como objetivo principal desarrollar un Sistema Informático que automatice las actividades que se requieren realizar para gestionar, organizar y controlar toda la información y la documentación técnico - legal requerida para la protección de los productos de Software Educativo en la Universidad de la Ciencias Informáticas. Acciones que se fundamentan a través de los procesos que se definen en la Estrategia de Trabajo diseñada en la universidad para lograr la protección de este tipo de productos. Para ello se realizó un estudio exhaustivo de los principales modelos, metodologías y estándares para el desarrollo de software, así como de las tendencias y tecnologías actuales, permitiendo este análisis seleccionar la metodología que apoya la solución del problema y las herramientas de desarrollo a emplear en la producción del software. Se realizó el análisis y diseño de la propuesta del sistema lográndose identificar las funcionalidades y la forma en que estas deben ser implementadas. Se desarrolló el sistema automatizado logrando abarcar en él todas las actividades que contribuyen a la correcta protección legal que se le debe conferir a los productos de Software Educativo, obteniéndose de esta forma una herramienta que facilita, viabiliza y apoya este proceso.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	6
1.1 INTRODUCCIÓN	6
1.2 ANÁLISIS DE LOS PRINCIPALES MODELOS, METODOLOGÍAS Y ESTÁNDARES PARA EL DESARROLLO DE SOFTWARE. SELECCIÓN DE LA METODOLOGÍA QUE APOYA LA SOLUCIÓN DEL PROBLEMA.....	6
1.2.1 <i>Rational Unified Process (RUP)</i>	7
1.2.2 <i>Extreme Programing (XP)</i>	7
1.2.3 <i>Microsoft Solution Framework (MSF)</i>	8
1.2.4 <i>Selección de la metodología a utilizar</i>	8
1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES. SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO.	9
1.3.1 <i>Principales lenguajes de programación. Selección del lenguaje a emplear en la implementación de la solución</i>	10
1.3.2 <i>Sistemas Gestores de Base de Datos (SGDB)</i>	13
1.4 CONCLUSIONES DEL CAPÍTULO.....	16
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....	17
2.1 INTRODUCCIÓN.	17
2.2 BREVE EXPOSICIÓN DE LA SITUACIÓN ACTUAL.....	17
2.3 OBJETO DE AUTOMATIZACIÓN.....	18
2.4 PROPUESTA DE SISTEMA.	23
2.5 MODELADO DE DOMINIO.....	25
2.5.1 <i>Glosario de Términos asociados al Dominio</i>	25
2.5.2 <i>Diagrama de Dominio</i>	27
2.6 REQUERIMIENTOS FUNCIONALES.	28
2.7 REQUERIMIENTOS NO FUNCIONALES.	32
2.8 MODELO DE CASOS DE USO DEL SISTEMA.....	33
2.8.1 <i>Diagrama de Casos de Uso del Sistema</i>	34
2.8.2 <i>Actores del Sistema</i>	35
2.8.3 <i>Descripción de Casos de Uso del Sistema</i>	35
2.9 CONCLUSIONES DEL CAPÍTULO.....	61
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	62
3.1 INTRODUCCIÓN.	62
3.2 DIAGRAMAS DE CLASES DEL ANÁLISIS.	62

3.3 MODELO DE DISEÑO.....	62
3.3.1 <i>Diagramas de Clases Web.....</i>	62
3.3.2 <i>Descripción de Clases del Diseño.....</i>	63
3.3.3 <i>Diagrama de Clases Persistentes.....</i>	78
3.3.4 <i>Diseño de la Base de Datos.....</i>	78
3.3.5 <i>Patrones de Diseño.....</i>	78
3.3.6 <i>Definiciones de diseño.....</i>	80
3.3.7 <i>Tratamiento de errores.....</i>	81
3.3.8 <i>Seguridad.....</i>	82
3.3.9 <i>Interfaz.....</i>	82
3.3.10 <i>Estándares de codificación.....</i>	83
3.3.11 <i>Concepción de la ayuda.....</i>	84
3.4 CONCLUSIONES DEL CAPÍTULO.....	84
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	85
4.1 INTRODUCCIÓN	85
4.2 MODELO DE DESPLIEGUE.....	85
4.3 MODELO DE IMPLEMENTACIÓN.....	86
4.4 MODELO DE PRUEBA.....	86
4.4.1 <i>Caso de Uso Gestionar Cliente.....</i>	86
4.4.2 <i>Caso de Uso Administrar Proyecto.....</i>	88
4.4.3 <i>Caso de Uso Administrar Producto.....</i>	90
4.4.4 <i>Caso de Uso Gestionar Contrato.....</i>	93
4.5 CONCLUSIONES DEL CAPÍTULO.....	95
CONCLUSIONES.....	96
RECOMENDACIONES	97
REFERENCIAS	98
BIBLIOGRAFÍA.....	100
GLOSARIO DE TÉRMINOS Y SIGLAS.....	101

INTRODUCCIÓN

El uso de los principios de la Propiedad Intelectual en el campo de la industria del software ha ganado gran importancia y se ha convertido con el paso de los años en un elemento fundamental para sus desarrolladores, quienes han tenido que adaptar y aplicar mecanismos importantes para la protección legal de un producto de este tipo, como consecuencia de las nuevas regulaciones devenidas de esta rama del derecho.

En Cuba los centros e instituciones que dedican esfuerzos al desarrollo de software, han tratado de establecer sus propias estrategias de producción, incluyendo en ellas acciones encaminadas a mejorar y facilitar la aplicación de la correcta protección que ofrecen los principios de la Propiedad Intelectual, apoyándose para ello en la definición de un conjunto de elementos que organicen, elaboren y gestionen todos los componentes legales que se requieren para alcanzar este objetivo.

Tal es el caso de la Universidad de las Ciencias Informáticas (UCI), que conjunto a su labor docente de formar profesionales altamente calificados en la rama de la informática, la producción de software y servicios informáticos, se dedica en gran medida al desarrollo y producción de software de diferentes tipos. Donde se ha elaborado la propuesta de una estrategia de trabajo técnico – legal para la línea de producción de Software Educativo, con el objetivo de lograr introducir elementos que garanticen la protección de este tipo de productos según las legislaciones vigentes.

Esta propuesta surge debido a que actualmente en la producción de Software Educativo no se gestiona toda la documentación legal requerida de cada uno de los productos desde su inicio, sino que se realiza a partir del momento en que los productos se catalogan como terminados, donde con esta variante de solución empleada para lograr la protección legal de los productos educativos de la universidad no se ha logrado hasta el momento realizar una gestión eficaz y eficiente, ya que se realizan acciones tardías en ese sentido y se omiten elementos esenciales que ocasionan a su vez que todas las producciones logradas hasta el momento en esta área se encuentren vulnerables ante los preceptos de esta rama jurídica.

De acuerdo a la situación existente, a las características propias de la producción y a las particularidades de los aspectos que se deben tener en cuenta para lograr la protección, se puede decir que el correcto funcionamiento de la puesta en práctica de la nueva estrategia de

trabajo técnico – legal para la protección de Software Educativo se puede ver afectado en gran medida por:

- El desconocimiento por parte de los desarrolladores sobre los términos y procedimientos legales en materia de Propiedad Intelectual asociados a los proyectos.
- La magnitud creciente de la producción que abarca la ejecución simultánea de muchas líneas de desarrollo con numerosos productos dentro de cada una de ellas. Lo que implica la obtención concurrente de gran cantidad de productos educativos de diferentes tipos y características.
- La rapidez con la que en muchas ocasiones se requiere desarrollar los productos inducidos por la inmediata y necesaria implantación de los mismos, por el cumplimiento en tiempo de las fechas de entregas establecidas.
- La necesidad de participación armónica en la producción industrial de gran cantidad de especialistas (Equipo de diseño instruccional, Equipo de diseño visual, Equipo de producción de medias y Equipo de ingeniería de software) donde cada cual aporta un componente que se integra a la solución completa. Esto involucra para cada producto obtenido un número grande de personas que pueden ser tanto del centro como de otra entidad. Lo que provoca que se torne más compleja la obtención de los datos (nombre, número de identidad y su desempeño o responsabilidades en la confección del software) necesarios para el registro del producto, de todos los involucrados en el desarrollo del software.
- La particularidad de la inclusión de contenidos (que pueden ser preexistentes) dentro de los productos de Software Educativo que a diferencia de otros productos de software son parte integral del producto y que son suministrados tanto por especialistas internos como externos a la institución sin un control de la cantidad y procedencia de los mismos, donde se requiere identificar en cada una de las fases de desarrollo, todos los componentes u obras que sean preexistentes con el objetivo de gestionar las licencias de uso correspondientes. Este proceso puede tardarse y tornarse engorroso sobre todo si existe gran cantidad de estas obras incluidas en el producto.
- La adquisición y control de los permisos o licencias de uso de cada una de las obras identificadas como preexistentes.

- El control de la información necesaria de los contratos con todas las entidades que solicitan servicios de producción de Software Educativo y de los subcontratos de servicios por parte de la universidad a otras instituciones.
- La confección y control de documentos técnicos legales requeridos para el registro.

Teniendo en cuenta lo anteriormente expuesto se puede considerar que para lograr el buen funcionamiento de los procesos asociados a la protección legal de los productos de Software Educativo se requiere de un control estricto y una correcta gestión de toda la información y documentación legal que se genera a partir del comienzo del desarrollo de cada uno de los productos de este tipo.

De ahí se puede establecer que el **problema** a solucionar consiste en: ¿Cómo contribuir al proceso de gestión legal utilizado para la protección de los productos de Software Educativo en la Universidad de las Ciencias Informáticas?

Por tanto el **objeto de estudio** de este trabajo es la producción de Software Educativo en la Universidad de las Ciencias Informáticas.

De ello se deriva que el **campo de acción** que abarca este trabajo, es el proceso de gestión legal de productos de Software Educativo en la Universidad de las Ciencias Informáticas.

El **objetivo general** de este trabajo será: desarrollar un sistema automatizado que permita gestionar, organizar y controlar toda la documentación legal requerida para la protección de los productos de Software Educativo en la Universidad de la Ciencias Informáticas.

Los **objetivos específicos** son:

- Realizar un estudio del arte asociado al dominio del problema.
- Realizar análisis y diseño del sistema.
- Desarrollar el sistema de gestión legal propuesto.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **Tareas**:

- Análisis del proceso de gestión legal para la protección de un producto de Software Educativo.
- Análisis y selección de las tecnologías y herramientas a emplear para el desarrollo del sistema.
- Selección de la Metodología de Análisis y Diseño de Sistemas Informáticos, que facilite la creación y garantice la calidad del sistema.
- Diseño de una Base de datos que soporte las funcionalidades del sistema.
- Realizar el análisis y diseño del sistema que se propone.
- Implementación de un sistema que facilite la gestión legal de los productos de Software Educativo en la UCI.

El trabajo está estructurado en: 4 capítulos, conclusiones, recomendaciones, bibliografía, anexos y glosario.

El **Capítulo 1** presenta un estudio de las tendencias actuales de desarrollo de software con el objetivo de realizar la selección de las herramientas adecuadas o idóneas para ser empleadas en la solución del problema. Se expone un análisis de las metodologías de desarrollo de software más comunes, las características esenciales de los principales lenguajes de programación empleados en el desarrollo de aplicaciones web, así como las particularidades y funcionalidades de los posibles sistemas gestores de bases de datos a emplear en la solución.

El **Capítulo 2** contiene una breve exposición de la situación existente en el ámbito de la gestión legal del software educativo donde se deja en evidencia la necesidad de un sistema que esté basado en la estrategia técnico-legal, y que complemente esta tarea. Se expone el objetivo por el cual se precisa de la automatización, explicándose para esto, las particularidades de la estrategia a usar y el por qué de su surgimiento, así como su vínculo con el sistema que se pretende desarrollar. Se realiza también una propuesta de sistema que abarca los procesos involucrados y las tareas que permitirá realizar el mismo. Se confecciona un modelo de dominio. Se enumeran los requisitos funcionales y no funcionales. Se presenta un diagrama de Casos de Uso, donde se identifican las relaciones de los actores con el sistema, realizándose también la descripción de los mismos.



El **Capítulo 3** presenta los diagramas de clases del análisis, con el objetivo de brindar una visión de fácil entendimiento acerca de los procesos involucrados en el Sistema. Se realiza un modelo de diseño, en el cual se explica la interacción entre los artefactos que intervienen en la Aplicación Web. Además se hace un estudio para determinar los patrones de diseño a utilizar y como implementar el tratamiento de errores, la seguridad, diseño de la interfaz y como estará concebida la ayuda que brindará el Sistema. Este capítulo es de vital importancia puesto que constituye un hito para el comienzo de la implementación del software.

En el **Capítulo 4** se muestra el diagrama de despliegue en el cual se representará la relación entre los nodos del sistema. También se presenta el modelo de implementación y sus diagramas de componentes. Por último se muestran las pruebas realizadas al sistema con el objetivo de corroborar el correcto funcionamiento del mismo.

CAPÍTULO 1 . FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

En este capítulo se realiza un estudio de las tendencias actuales de desarrollo de software con el propósito de realizar una selección de las herramientas más eficientes en el desarrollo de la aplicación que dará solución al problema.

Para lograr los objetivos de esta fase, se expone un análisis de las metodologías de desarrollo de software más comúnmente utilizadas, así como las características esenciales de los principales lenguajes de programación empleados en el desarrollo de aplicaciones Web y en ese mismo sentido se analizan las particularidades y funcionalidades de los posibles sistemas gestores de bases de datos a emplear en la solución del problema propuesto.

1.2 Análisis de los principales modelos, metodologías y estándares para el desarrollo de software. Selección de la metodología que apoya la solución del problema.

A nivel internacional las instituciones y empresas dedicadas a la industria del software emplean en su actividad de desarrollo, modelos, metodologías o procedimientos estándares para desarrollar, instalar y mantener un producto de este tipo. Dichos modelos permiten uniformar la filosofía de trabajo, en aras de lograr la obtención de un software de calidad, y a la vez elevar la productividad para así competir y poder acceder a las oportunidades del mercado mundial (ANAYA 2005).

Para lograr un correcto desarrollo de software, estas metodologías establecen un conjunto de actividades que definen cómo se debe hacer el software, quién debe hacer cada actividad, cuándo hacerla y qué se debe hacer.

En la actualidad existen gran cantidad de metodologías orientadas al proceso de desarrollo de software, entre las más significativas se pueden mencionar el Rational Unified Process (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF).

1.2.1 Rational Unified Process (RUP).

El Rational Unified Process (RUP) es un proceso iterativo e incremental que define quién, qué, cómo y cuándo deben hacerse las cosas. Está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes, utilizando para ello el Lenguaje de Modelado Unificado (UML, Unified Modeling Language) el que define técnicas de análisis y diseño que ayudan a la confección de una solución sólida de software (JACOBSON 2007).

RUP se caracteriza por ser **Dirigido por casos de uso** donde los casos de uso definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio. Es **Centrado en la Arquitectura**, característica que brinda una visión completa del sistema, se describen los procesos del negocio que son más importantes, para comprenderlo, desarrollarlo y producirlo de una forma eficaz. **Iterativo e Incremental** donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo (JACOBSON 2007).

De forma general RUP divide en 4 fases el desarrollo del software, estas son: **inicio** la que define el modelo del negocio, el alcance y los límites del proyecto, **elaboración** en la cual se define, valida y cimenta la arquitectura, **construcción** donde se desarrolla el producto y **transición** la encargada de poner el producto en manos de los usuarios. (JACOBSON 2007)

1.2.2 Extreme Programming (XP).

Extreme Programming es una metodología de desarrollo utilizada para proyectos de corto plazo y pequeño equipo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (SÁNCHEZ 2007).

Se basa en tres características principales, las **Pruebas Unitarias** que no son más que pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran ocurrir.

La Refabricación que establece la reutilización de código, para lo cual se crean patrones o modelos estándares, haciéndolo más flexible al cambio, y la **Programación en Pares**, que consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo, donde cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento (SÁNCHEZ 2007).

Presenta como principal propósito la satisfacción del cliente, proporcionándole lo que necesita y cuando lo necesita, además de potenciar al máximo el trabajo en grupo. Dicha metodología solo puede funcionar con programadores experimentados en la materia, y solo para pequeños grupos de estos. Está orientada más a las personas que a los procesos y es pobre en la documentación (SÁNCHEZ 2007).

1.2.3 Microsoft Solution Framework (MSF).

La metodología de desarrollo de software MSF es flexible y está interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos. Se centra en los modelos de proceso y de equipo (CANSECO 2007).

Posee cuatro características principales, **la adaptabilidad** ya que es adaptable a un proyecto de cualquier magnitud. **La escalabilidad**, pudiéndose organizar equipos de tres o cuatro personas, así como también proyectos que requieren 50 personas o más. **La flexibilidad**, ya que se puede utilizar en el ambiente de cualquier cliente, y el hecho de ser una **tecnología agnóstica** ya que puede ser empleada para desarrollar soluciones basadas sobre cualquier tecnología (CANSECO 2007).

MSF consta de las siguientes fases para su desarrollo: **Visión**, donde se hace una descripción general de las metas y restricciones del proyecto. Aquí se identifican las tareas y entregables. **Planeación** es en la que el equipo determina qué desarrollar y planea cómo crear la solución. El equipo prepara la especificación funcional, crea el diseño de la solución y prepara los planes de trabajo, estimaciones de costos, y agenda el cumplimiento de los entregables. El **Desarrollo**, donde se crea el código que implementa la solución y su documentación. **Estabilización**, es en la que el equipo integra, carga y realiza pruebas de la solución. **Implementación**, donde el equipo implementa la solución tecnológica y sus componentes, estabiliza la implementación, transfiere el proyecto a producción y soporte, y obtiene la aprobación final del cliente sobre el proyecto (CANSECO 2007).

1.2.4 Selección de la metodología a utilizar.

En correspondencia con el análisis realizado anteriormente sobre las metodologías seleccionadas, se puede evidenciar la presencia de diferentes características para cada una de ellas, y la particularidad de poder ser empleadas para diferentes situaciones respectivamente.

La metodología XP está destinada a proyectos de corto plazo, se basa en iteraciones pequeñas y está dirigida fundamentalmente a los clientes. MSF por su parte es una metodología adaptable a proyectos de cualquier dimensión, sin embargo, no basa su funcionamiento en la documentación del software sino en el control de entregables por parte de los desarrolladores. RUP es perfectamente idónea para proyectos de largo plazo, aunque es caracterizada por ser una metodología compleja.

Teniendo en cuenta las particularidades de cada una, se ha seleccionado como apoyo en el desarrollo de la aplicación propuesta la metodología RUP, la cual basa su trabajo fundamentalmente en la documentación del software y expone un conjunto de actividades que están orientadas a visualizar, especificar, construir, documentar y comunicar los artefactos necesarios para el desarrollo de un software de calidad, presentando una exhaustiva definición de artefactos para ello. Además utiliza el lenguaje unificado de modelado (UML) una herramienta muy potente cuya utilización de diagramas y gráficos brindan una mejor perspectiva de lo que se quiere.

1.3 Tendencias y tecnologías actuales. Selección de las herramientas de desarrollo.

Para dar solución a la problemática planteada se ha propuesto desarrollar una Aplicación Web, según la petición realizada por los clientes debido a que la mayoría de las aplicaciones empleadas por ellos en su trabajo de producción son de este tipo y por otra parte teniendo en cuenta las grandes funcionalidades y posibilidades de implementación que brindan las herramientas de desarrollo en esta área.

Este tipo de aplicaciones presenta como principal peculiaridad el empleo del modelo cliente/servidor, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes (VALLE 2007).

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 usuario. El programa cliente gestiona la comunicación con el servidor y ofrece las herramientas necesarias para poder trabajar con dicho servidor Web (VALLE 2007).

El servidor Web es un pilar importante en esta tecnología, puesto que es el programa que corre en una máquina central escuchando las peticiones de los clientes y respondiendo a las mismas en forma de una página que es interpretada por éste y mostrada al usuario de una manera entendible para él. Los más conocidos y utilizados en la actualidad son Apache y el IIS (Internet Information Server).

El IIS es un software privativo que trabaja solo sobre Windows, por lo que en la solución propuesta se empleará el Apache, éste por su parte es un servidor que presenta un sin número de características que lo hacen estar entre los primeros en este ámbito. Es flexible, rápido y eficiente, puede ser ejecutado en cualquier plataforma y adaptado a diferentes entornos y necesidades. Se desarrolla de forma abierta y gracias a que es modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (como se expondrá más adelante).

Para poder seleccionar las herramientas a emplear en la implementación de la aplicación Web, se realizará un estudio de las principales tecnologías, lenguajes de programación y gestores de bases de datos.

1.3.1 Principales lenguajes de programación. Selección del lenguaje a emplear en la implementación de la solución.

Hoy en día existen diferentes lenguajes de programación que permiten desarrollar una aplicación de este tipo, los cuales se dividen en dos grupos fundamentales que reconoce la arquitectura Cliente/Servidor: Los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor más comúnmente utilizados por los programadores podemos encontrar: el Personal Home Page (PHP), Practical Extracting and Reporting Language (PERL), Active Server Pages (ASP). Estos lenguajes se caracterizan de forma general por desarrollar la lógica del negocio dentro del Servidor, de realizar el acceso y la actualización a Base de Datos, así como del tratamiento de la información.

Del lado del Cliente tenemos: JavaScript y al Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores permitiendo una mayor flexibilidad en la interfaz.

1.3.1.1 Practical Extracting and Reporting Language (PERL).

Perl es un lenguaje de programación de scripts que está disponible en múltiples plataformas y sistemas operativos. El desarrollo de aplicaciones en este lenguaje es muy rápido y se pueden incorporar a cualquier conjunto de instrucciones de él, una colección enorme de módulos que están favorables para su utilización.

Perl es gratuito, mucho más que eso, es “Software Libre”, esto quiere decir que el código fuente está disponible para que cualquiera lo pueda ver, utilizar o modificar, brindándole de esta forma al programador mucha libertad para que haga el programa como desee. La libertad que se le otorga al programador puede significar que el resultado sea un programa ilegible, si no se escribe con cuidado puede llegar a ser difícil de leer.

Perl trae consigo una serie de módulos DBD (Controlador de base de datos) que manejan los detalles de acceso a diferentes bases de datos (HIETANIEMI 2007).

Es un lenguaje que puede llegar a provocar que algunas aplicaciones sean lentas, sobre todo en el empleo de programación a bajo nivel o corriendo modelos numéricos de cálculo intensivo (HIETANIEMI 2007).

Por otra parte los programas desarrollados en este lenguaje no se pueden compilar. Utiliza muchos recursos de la máquina, lo que provoca que no sea tan ligero como otros lenguajes (HIETANIEMI 2007).

1.3.1.2 Active Server Pages (ASP).

Es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma NET de Microsoft, de lo cual se deduce que es un lenguaje privativo y de un alto costo. Este soporta acceso a bases de datos, trabajo con archivos y carpetas, envío de emails, paginación de resultados, procesado de formularios, y muchas otras opciones (WALTER 2006).

ASP es un tanto desorganizado. En una página ASP podemos incluir casi todo: HTML plano, código de scripting, texto. No hay una distinción formal entre el contenido de una página y su comportamiento. ASP.NET impone un cierto orden sobre el modelo de programación estándar ASP (WALTER 2006).

1.3.1.3 Personal Home Page (PHP)

PHP es un lenguaje script para el desarrollo de páginas Web dinámicas del lado del servidor, cuyo código se puede implementar por separado del HTML y unirlo a través del motor de plantilla SMARTY, lo que hace que el trabajo del diseñador gráfico y el del programador sean realizados independientemente. Es Open Source (código abierto) y funciona en todas las plataformas que soporten Apache, es el más popular y extendido en la Web. (VÁZQUEZ 2007)

Este es un sistema robusto y estable donde se pueden hacer grandes cosas con pocas líneas de código, lo que hace que merezca la pena utilizarlo. El código de este sistema es mucho más legible que el de PERL y muy sencillo de aprender, además de que viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor. Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos, con un excelente soporte. (VÁZQUEZ 2007)

PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz y está completamente escrito en C, por lo que se ejecuta rápidamente utilizando poca memoria. (VÁZQUEZ 2007)

Este lenguaje soporta en cierta medida la programación orientada a objeto (Clases y herencia) y también realiza de forma automática el análisis léxico para recoger las variables que se pasan en la dirección, lo cual libra al usuario de tener que separar las variables y sus valores. La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente, de forma que se puede evitar que se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.

Sin embargo, todo el trabajo lo realiza el servidor y no delega al cliente, por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número y la orientación a objetos es aún muy deficiente para aplicaciones grandes.

1.3.1.4 Fundamentación de la selección del lenguaje a utilizar.

Perl es un lenguaje de programación Web libre que brinda un sin número de características interesantes, sin embargo, consume muchos recursos de la máquina y se torna lento en aplicaciones de bajo nivel. ASP por su parte es privativo lo que justifica su enorme precio. PHP pertenece al grupo de programas de código abierto y libre.

Se ha seleccionado como lenguaje a utilizar en la implementación de la aplicación Web, al PHP. Este es un lenguaje gratuito, multiplataforma y debido a sus características hace posible que el cliente interactúe con una página rápida, eficiente y segura, capaz de mostrar y procesar información.

Unido a lo anterior, se puede mencionar que es muy sencillo y legible, y cuenta con una amplia librería de funciones que permiten hacer cualquier tipo de operación, como trabajo con archivos y carpetas, procesamiento de formularios, paginación de resultados y muchas otras opciones, lo cual concede la posibilidad de realizar un producto de acuerdo a las necesidades del usuario.

1.3.2 Sistemas Gestores de Base de Datos (SGDB).

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. En los textos que tratan este tema, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y Data Base Management Sistema, según su expresión inglesa. (WIKIMEDIA FOUNDATION 2007)

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. Un SGBD permite: la Independencia entre los datos y los programas de aplicación, minimizar las redundancias, integrar y sincronizar las bases de datos, garantiza la integridad, la seguridad y protección de los datos, facilita la de manipulación de la información y permite un control centralizado. (WIKIMEDIA FOUNDATION 2007)

Entre los SGBD más comunes se encuentra el Oracle, el Microsoft SQL Server, el MySQL y el PostgreSQL, a continuación se expondrán las características fundamentales de cada uno de ellos, lo que permitirá la selección del que se va a utilizar en la implementación de la Aplicación Web.

1.3.2.1 PostgreSQL.

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe, es además la alternativa más cercana a MySQL cuando se precisa de operaciones avanzadas como transacciones, procedimientos almacenados, vistas, o cuando se precisa de una base de datos que soporte gran cantidad de información. (GONZÁLEZ 2007)

Es el servidor más utilizado por todos aquellos programadores que realizan aplicaciones cliente servidor, complejas o críticas. (GONZÁLEZ 2007)

Es muy utilizado actualmente y es una alternativa económica a SQL Server, pues su costo es menor y tiene similares prestaciones. Este se puede utilizar sobre cualquier sistema operativo, característica que lo pone por encima de SQL server y al parejo con MySQL.

La mayor limitación que posee viene dada por su velocidad ya que se torna un tanto lento, en la realización de operaciones elementales de los sistemas gestores de base de datos. (GONZÁLEZ 2007)

Además consume muchos recursos por lo que puede sobrecargar el sistema, tiene un límite de tamaño para los registros, y aunque es compatible con PHP tiene muy pocos comandos definidos para el tratamiento de este. (CASTELLANO 2007)

1.3.2.2 SQL Server.

SQL Server es un sistema de base de datos muy completo y potente. Posee una gran velocidad y soporta un volumen de datos muy grande. También, presenta mecanismos que le permiten realizar sentencias complicadas, lo que lo hace perfectamente adecuado para aplicaciones críticas y con cualquier grado de complejidad. Por otro lado reserva una parte de la base de datos para guardar el registro de transacciones con los comandos pendientes, lo que asegura que independientemente de que el programador use o no transacciones en su código, en ningún caso la base de datos quedaría en un estado inconsistente debido a una ejecución parcial de comandos. Además ofrece características avanzadas orientadas a la integridad de la base de datos. Posee el inconveniente que corre solamente sobre una plataforma, Windows, además de ser un software privativo con un alto costo. (MSDN 2007)

1.3.2.3 MySQL.

MySQL por su parte es un servidor de base de datos que ostenta la cualidad de poseer un rápido nivel de procesamiento. Es el más utilizado cuando se emplean lenguajes de programación como PHP y Perl, y cabe mencionar su gran afinación principalmente con PHP (SCHUMACHER 2007).

MySQL es un software libre, confiable y fácil de usar, es multiplataforma, multiusuario y permite elaborar consultas con el robusto SQL, consume muy pocos recursos tanto del CPU como de memoria, presenta mejoras en utilidades de administración (backup, recuperación de errores, etc.), no hay límites en el tamaño de los registros, tiene como una de sus principales ventajas la velocidad en la lectura de datos (SCHUMACHER 2007).

En recientes versiones de MySQL (la versión 4 y la 5) se incluyen un conjunto de facilidades que no tenía en versiones anteriores como el uso de procedimientos almacenados, bloqueo de registros, integridad referencial, transacciones, entre otros, lo que aumenta aún más sus ventajas (SCHUMACHER 2007).

1.3.2.4 Oracle.

Es considerado el SGBD más completo que existe. Sus características más destacadas son el soporte de transacciones, su gran estabilidad y seguridad, su escalabilidad, así como que es un sistema multiplataforma, entre otras ventajas. "Su mayor defecto es el "enorme precio", que es de varios miles de euros (según versiones y licencias) (MASIP 2007).

Un aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministros de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios (MASIP 2007).

Hasta hace poco su dominio en el mercado de los servidores de bases de datos empresariales era casi total, pero recientemente está sufriendo la competencia del SQL Server de Microsoft y de la oferta de otros SGBD libres como MySQL y Postgre (MASIP 2007).

1.3.2.5 Fundamentación de la selección de la Base de Datos a utilizar.

Después de haber realizado un análisis completo de los diferentes tipos de bases de datos más utilizados en la actualidad y teniendo en cuenta los aspectos más significativos por los cuales se puede comparar, se llega a la conclusión de que el más idóneo para llevar a cabo los objetivos es el MySQL.

Poniendo especial atención a las restricciones del cliente, es un software libre y además es utilizable en cualquier plataforma. Por otro lado y comparándolo con los otros tipos de bases de datos su mayor adversario es SQL Server el cual es software propietario. Postgre por su parte a pesar de su robustez se torna un poco lento. Además MySQL posee una gran afinidad con PHP y es el más eficiente para aplicaciones no tan complejas como la que se pretende desarrollar.

1.4 Conclusiones del Capítulo.

Teniendo en cuenta las consideraciones anteriormente expuestas se puede indicar que para el desarrollo de la Aplicación Web que apoyará, gestionará y controlará la información y documentación legal requerida para la protección de los productos de Software Educativo, se utilizará como lenguaje de programación del lado del servidor el PHP, apoyándose en el uso del motor de plantillas SMARTY, y el JavaScript del lado del cliente para lograr una mejor interactividad con el usuario en el navegador. Como Sistema de Gestión de Bases de Datos se empleará el MySQL, utilizando la librería ADOdb que proporciona el lenguaje PHP para el acceso a los datos, y el Apache como servidor Web. Para realizar el análisis y diseño del sistema se empleará la metodología de desarrollo RUP.

CAPÍTULO **2**. CARACTERÍSTICAS DEL SISTEMA.

2.1 Introducción.

En este capítulo se realiza la descripción de la situación actual referente a la gestión legal del software educativo en la UCI, se define el objeto de automatización explicándose para ello, la estrategia de protección técnico - legal que fundamenta dicho proceso, así como su vínculo con el sistema que se propone desarrollar. Se realiza además la propuesta de las características que presentará el sistema informático que dará respuesta a la problemática existente donde se interrelacionan los procesos involucrados en la estrategia con las funcionalidades que permitirá realizar el mismo. Por otra parte se confecciona el modelo de dominio que agrupa los conceptos fundamentales en función de una mejor captura de requisitos. Se definen los requisitos funcionales y no funcionales y se confecciona un diagrama de casos de uso, donde se identifican las relaciones de los actores con el sistema.

2.2 Breve exposición de la situación actual.

Para poder lograr la protección legal que requieren los productos de software educativo a través de las normativas que establecen las legislaciones de la rama del Derecho de la Propiedad Intelectual se diseñó en la universidad una estrategia o metodología que pauta la forma en que se debe proceder para lograr este propósito, la cual aun no ha podido ser utilizada en el proceso de producción de este tipo de software, producto a que:

- No se posee un control estricto de la información manipulada en virtud del desarrollo de los proyectos.
- La metodología propone la elaboración de una gran cantidad de documentación técnico – legal por cada tipo de producto que se desarrolle.
- Existe desconocimiento por parte de los desarrolladores en cuanto a la protección legal que proporciona la Propiedad Intelectual a este tipo de productos.
- No se posee un correcto dominio del funcionamiento de la metodología.

- Confección tardía de documentos legales requeridos.

Además, de la gran cantidad de información que este genera, existe un continuo y creciente desarrollo simultáneo de varias líneas de producción con gran cantidad de productos dentro de cada una de ellas, la participación de diferentes especialistas en la solución, la rapidez con la que en muchas ocasiones se requiere que se desarrollen los productos inducidos por la inmediata y necesaria implantación de los mismos, por el cumplimiento en tiempo de las fechas de entregas establecidas, el control de los documentos requeridos para el registro por cada producto y en los procesos de contratación y subcontratación con cada entidad cliente, así como el control de los datos de todos los involucrados en el desarrollo.

Producto a esta situación es que se requiere de un sistema automatizado que apoye la correcta ejecución, gestión y control de este proceso de protección legal de los productos de software educativo a partir de las definiciones que propone la estrategia diseñada para este propósito.

2.3 Objeto de automatización.

En correspondencia con las disposiciones de las legislaciones nacionales e internacionales en materia de Propiedad Intelectual se realizó la propuesta de estrategia de trabajo técnico – legal para la protección de Software Educativo, la cual está compuesta por un conjunto de subprocesos que están vinculados e insertados durante el ciclo de desarrollo de este tipo de producto.

Para poder tener una mayor comprensión de los aspectos que plantea esta estrategia se realizará una pequeña exposición del flujo de trabajo empleado para el desarrollo de Software Educativo sobre el cual se apoya la propuesta técnico - legal.

Con el propósito de facilitar el entendimiento del sistema, se ha diseñado un sencillo gráfico arborescente (Figura 2.1) que muestra los procesos asociados al ciclo de desarrollo de este tipo de producto según las definiciones realizadas por la Dirección de Producción No. 2, la cual asume la asesoría metodológica y el control de los procesos asociados a la producción de Software Educativo de la Universidad.

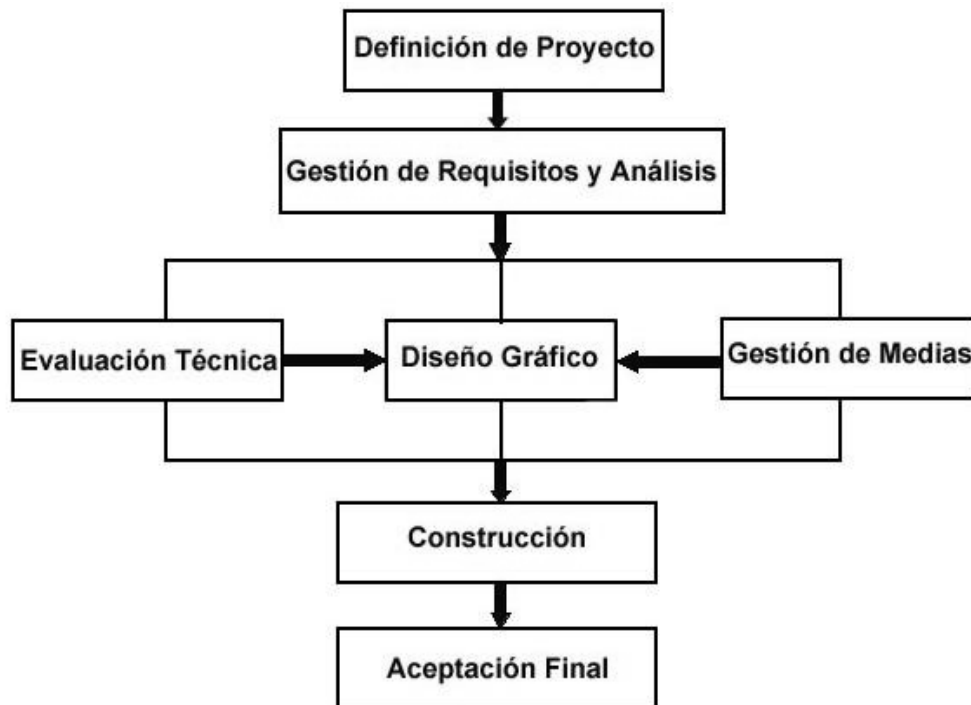


Figura 2.1 Flujo de Trabajo del proceso de producción de Software Educativo (MARTÍNEZ, 2006).

Aquí se identifican siete procesos como parte del flujo de trabajo general: Definición del proyecto, Gestión de requisitos y análisis, Evaluación técnica, Gestión de medias, Gestión de diseño gráfico y Construcción. A este flujo de trabajo se le van insertando a medida que transcurre su ejecución, los subprocesos asociados a la gestión legal en dependencia del proceso al que estén asociados.

De forma general y sin adentrarse en las particularidades de este flujo se explicará la esencia de cada uno de los procesos que la componen, puesto que no constituye el tema de especial atención del presente trabajo, pero sí, forma la base integral sobre la cual se inserta la estrategia de gestión legal.

La primera etapa, **Definición del proyecto**: Describe toda la etapa de concepción inicial del producto. Se traza la estrategia de trabajo durante el desarrollo del proyecto para el equipo de proyecto, incluyendo trabajo con el cliente.

Gestión de requisitos y análisis: Se definen las funcionalidades que tendrá el producto y se realiza la documentación requerida para la continuidad de los siguientes procesos.

Evaluación técnica: Se efectúan las especificaciones de los productos a elaborar, se realiza una recomendación de la arquitectura a usar para la producción de Software Educativo, así como de la arquitectura organizativa para acometer la producción.

Gestión de medias: Comprende los procesos de búsqueda o producción de determinados recursos audiovisuales en función de los requisitos establecidos en el segundo proceso.

Gestión de diseño gráfico: Comprende los procesos de concepción realización del diseño gráfico de un producto de Software Educativo en función de los requisitos establecidos en la documentación obtenida en el segundo proceso.

Construcción: Se realiza la implementación del producto, abarcando las etapas de revisión y corrección de errores.

Aceptación final del producto: Entrega y aceptación del producto por parte del cliente.

Con estas precisiones, se pueden enmarcar con mayor claridad a los subprocesos que están inmersos en el proceso de gestión legal para la protección de software educativo. El siguiente gráfico (Figura 2.2) muestra la forma en que van dispuestos.

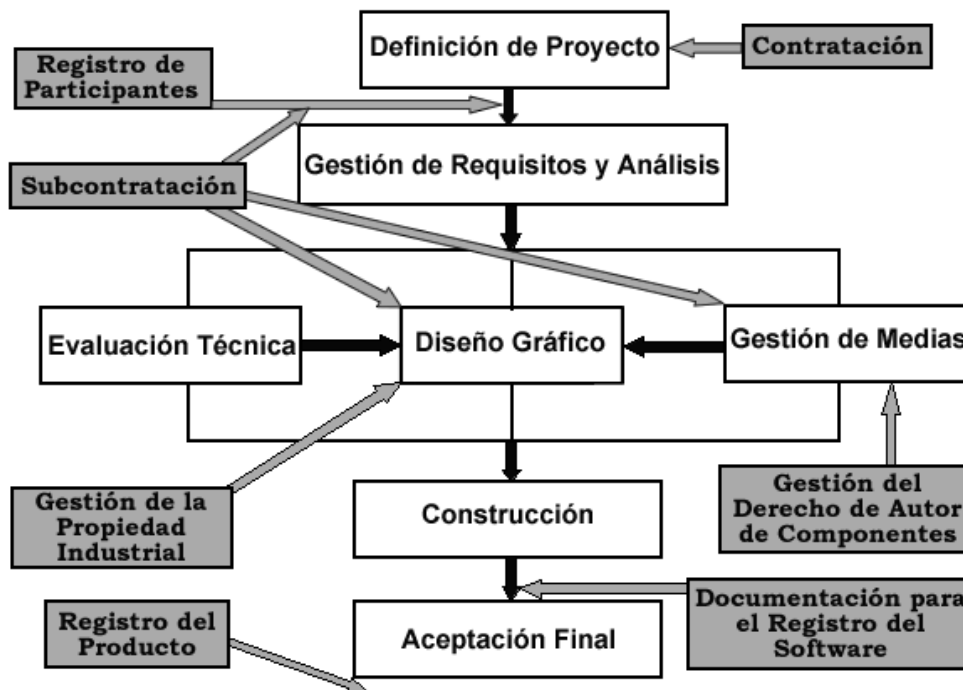


Figura 2.2 Estrategia Técnico-Legal. (BÁEZ, 2006)

Durante la fase Definición del Proyecto se propone la entrega por parte del cliente del documento de solicitud de ejecución del **proyecto**, donde se especifican los datos necesarios de la institución **cliente** (nombre de la empresa, tipo de empresa (nacional o extranjera), teléfono, fax, siglas, etc.), una descripción del problema a resolver y las características de los **productos** a desarrollar (arquitectura de desarrollo, restricciones y tiempo de producción estimado, otros) y el guión de contenido (documento que describe la secuencia detallada de contenidos que debe contener un producto) este último, de no estar definido en un momento inicial, puede ser entregado o elaborado antes de culminar este proceso. Después de estar establecidas las condiciones y especificaciones iniciales del proyecto se selecciona el personal que estará involucrado en el desarrollo del producto para lo cual se requiere de un estricto control de sus datos, registrándose de cada **integrante**, (el nombre, la responsabilidad, la institución a la que pertenece y los datos de contacto) y se pone en práctica un compromiso de ética para establecer determinadas bases de confidencialidad. Por otro lado se define el cronograma general e inicial del proyecto (documento que contiene las actividades y tareas a realizar especificando en cada caso, fechas, recursos necesarios, participantes y responsables).

Una vez culminada esta etapa, se tiene una visión general de todas las características esenciales de la producción del software solicitado (tipo de producto, destinatarios, funciones del producto, tiempo de desarrollo, tipo de especialistas involucrados, costo y precio, entre otras), por lo que se le da paso al establecimiento de un **contrato** que garantice la legalidad de las condiciones, compromisos y responsabilidades entre las partes involucradas en el desarrollo del producto educativo. Para esto se precisa de documentos **anexos** y **suplementarios**, que garanticen la no omisión del registro de: los autores y titulares del software en cuestión, los derechos morales y patrimoniales haciendo gran énfasis en los derechos de reproducción, transformación, distribución, comunicación pública, las modalidades de explotación, versiones sucesivas, limitaciones contenidas en los derechos, la titularidad y los derechos sobre cada una de los **recursos audiovisuales** de **nueva creación** que será necesario confeccionar, responsabilidad de la gestión de las **licencias de uso** de **obras preexistentes**, responsabilidad de la gestión de las licencias de uso de las herramientas a emplear y otras disposiciones ligadas al desarrollo del producto.

Como particularidad esencial de un software educativo se encuentra la inclusión dentro del producto de **obras audiovisuales** (textos, imágenes, videos, animaciones, música, etc.) que pueden ser de dos tipos: **preexistentes** (que ya existen y fueron realizadas por otros autores),

o de **nueva creación** (las que son realizadas para ese producto en especial). Estas últimas, pueden ser desarrolladas por diferentes tipos de especialistas, diseñadores, músicos, animadores, ilustradores y otros. En la mayoría de los casos estos especialistas pertenecen a nuestra universidad, pero en ocasiones se requiere de la ayuda de otras instituciones (el personal especializado en estas áreas no alcanza para poder desarrollar las demandas de la gran cantidad de productos que se realizan), para lo cual se realiza la **subcontratación** de servicios, de la cual, al igual que para el contrato se registra y controla la documentación del mismo.

En el caso de las **obras preexistentes** para poder usarlas en el software educativo se requiere de la gestión de los permisos o **licencias de uso** de los mismos, proceso que se cataloga también como subcontratación y de las cuales se recogen una serie de datos necesarios.

Durante la etapa de Diseño Gráfico se puede obtener el **diseño** del producto o colección de productos y en algunas ocasiones se realiza también un diseño de **marca** de colección de productos u otros componentes visuales, en dependencia de la originalidad de estos diseños se decidirá o no por parte del personal el **registro** en las instituciones legales requeridas, para lo cual se debe elaborar el Manual de Identidad Visual y gestionar los datos de los autores.

Durante la ejecución del proceso gestión de medias se debe tener un control estricto de las obras (nombre, autor, procedencia, etc.) tanto preexistente como de nueva creación, y al igual que al diseño en dependencia de la originalidad que posean estas producciones se decidirá si se procederá a realizar el **registro legal** de cada una de ellas. Donde para ello se requiere de determinada documentación en dependencia del tipo de obra de que se trate.

Una vez terminado el **producto** se procede a la elaboración de la documentación requerida para el **registro** del software final (**Manual de usuario** y **Ficha Técnica**), de los que se requerirá tener un control para no mezclarlo ni confundirlo con la documentación de ningún otro producto.

Para el proceso de registro del software se requerirá además de la documentación anteriormente mencionada, las **licencias de uso** de **obras preexistentes** incluidas en la solución del Software, así como un ejemplar del producto.

2.4 Propuesta de sistema.

Con el objetivo de complementar la estrategia técnico legal propuesta para la protección de los software educativos creados en la Universidad de las Ciencias Informáticas, se ha decidido desarrollar un sistema que incorpore los elementos que brinda dicha estrategia y que garantice un correcto funcionamiento de la misma. El mismo brindará amplias facilidades para almacenar y manejar la información relacionada con los productos en cuestión. A continuación se expondrán los diferentes servicios que ofrecerá dicho sistema.

Para el trabajo con el sistema se han definido tres roles, o sea un usuario que se puede comportar como **administrador**, el cual tendrá acceso a todas las funcionalidades del sistema, o también como **líder de proyecto** o **líder de producto** los cuales tendrán acceso a determinadas funcionalidades en dependencia de su rol y solo a los proyectos y productos de los cuales ellos sean líderes respectivamente. La información de dichos usuarios se registrará y controlará en la base de datos del sistema.

El sistema permitirá insertar los datos referentes a la institución **cliente** que solicita la realización del software educativo. Una vez insertados, consultarlos y modificarlos en caso necesario, así como, una vez culminado el producto y consigo la relación con la universidad eliminarlos del sistema. Estas acciones solo podrán ser realizadas por el administrador.

El sistema permitirá insertar además los datos del **proyecto**, acción que realizará el administrador, y una vez hecha la inserción, tanto el administrador como el líder de proyecto podrán modificarlos y consultarlos de ser necesario. Cuando el proyecto se culmine y no sea preciso tenerlo registrado el administrador podrá eliminarlo.

El sistema permitirá insertar los datos del **producto**, acción que podrá realizar el administrador o el líder de proyecto con los productos que pertenezcan al proyecto del cual él es líder, compartiendo también la acción de eliminarlos una vez culminado el mismo. Se podrá modificar y consultar los datos de los productos, acciones a las cuales también tendrá acceso el líder de producto.

El sistema permitirá insertar los datos del **contrato**, y una vez insertados, modificarlos, consultarlos o eliminarlos cuando sea preciso. Así como, incorporar o eliminar los documentos **anexos** o **suplementarios** del contrato en cuestión. Estas acciones solo las podrá realizar el administrador.

El sistema permitirá además insertar, modificar, consultar y eliminar los datos de los **integrantes** del proyecto que tomarán parte en la realización de cada producto y de los cuales es necesario tener un estricto control. Así como realizar estas mismas acciones, esta vez respecto a la **ficha técnica** de dicho producto, además de generar un documento con los datos de la misma. A estas funcionalidades podrán tener acceso el administrador, el líder de dicho proyecto y el líder del producto.

Además permitirá realizar acciones de inserción, modificación, consulta y eliminación de los datos referentes a las **solicitudes de servicio** y a las **licencias de uso**, necesarias para el desarrollo de obras de nueva creación o adquisición de permisos para el uso de obras preexistentes respectivamente. Así como subir la proforma de la solicitud de servicio y el documento licencia de uso. A estas funcionalidades tendrán acceso el administrador, el líder de dicho proyecto y el líder del producto.

El sistema también deberá gestionar tanto las **obras de nueva creación** como las **obras preexistentes** permitiendo insertar, modificar, consultar y eliminar dichas obras. Y para el caso de las de obras de nueva creación, generar una solicitud de registro e inscribirla como registrada en el Centro Nacional del Derecho de Autor (CENDA). A estas funcionalidades podrán tener acceso el administrador, el líder de dicho proyecto y el líder del producto.

El sistema permitirá insertar, modificar, consultar y eliminar el **diseño** y la **marca** correspondiente a cada producto. Y para ambos generar una solicitud de registro e inscribirlas como registradas en el CENDA. Así como incorporar o eliminar una copia de la marca. A estas funcionalidades podrán tener acceso el administrador, el líder de dicho proyecto y el líder del producto.

Una vez terminado el **producto** generar una solicitud de registro del mismo y emitir un informe para su registro, además de inscribirlo como registrado en el CENDA.

Con el objetivo de brindar una mayor información acerca de los productos, el sistema le permitirá la gestión de **reportes**, al administrador, mostrándole la cantidad de productos registrados, los no registrados y el total de ellos, así como los productos por proyecto o por proyectos y clientes. Y también mostrar las obras audiovisuales de cada producto de acuerdo a su clasificación (preexistente o de nueva creación).

Por último y no menos importante permitirá gestionar **usuarios**. El administrador podrá insertar, modificar, consultar y eliminar los datos de los usuarios, así como restringir su acceso a cada funcionalidad y su **autenticación**.

2.5 Modelado de dominio.

Debido a que la estrategia de trabajo técnico legal aún no está siendo empleada en la producción de Software Educativo, no se puede considerar que exista un negocio bien definido y consistente que permita determinar las responsabilidades, entidades y funcionamiento de este proceso. Por tal motivo se ha representado los conceptos que se manejan en la situación existente a través de un Modelo de Dominio el cual brinda una mayor comprensión de los eventos que suceden en el entorno en el que se trabaja.

2.5.1 Glosario de Términos asociados al Dominio.

Para obtener una mejor comprensión del modelo que se plantea es necesario utilizar un glosario de términos en el cual se identifican los conceptos a utilizar.

Universidad: es la institución que se encarga del desarrollo del software educativo solicitado por el cliente.

Cliente: institución que solicita la elaboración de uno o varios productos con el fin de satisfacer una necesidad propia. Dicha institución puede ser nacional o extranjera.

Proyecto: es la herramienta o instrumento que recopila, crea y analiza en forma sistemática un conjunto de datos de los productos, para la obtención de resultados esperados y la organización del entorno de trabajo.

Producto: es el resultado esperado de todo el desarrollo del proyecto, donde se generan obras, marcas, diseños, etc.

Integrante: persona que participa en la elaboración de uno o varios productos.

Contrato: es el proceso donde se formaliza la relación entre las entidades para el desarrollo de un Proyecto donde se contemplan todas las especificaciones referentes a la relación, derechos y deberes de las entidades participantes.

Subcontratación: proceso de contratación llevado a cabo para la solicitud de un servicio u obtención de una Licencia de Uso, para realizar alguna tarea específica o utilizar algún elemento necesario para el desarrollo del producto.

Recursos Audiovisuales: obras utilizadas para la elaboración de un producto. Medios de comunicación que se utilizan en la presentación de la información, (sonido, video, imagen, etc.). Pueden ser obras preexistentes o de nueva creación.

Obra de Nueva Creación: recurso audiovisual creado específicamente durante el desarrollo del producto y que pasará a formar parte del conjunto de medios de comunicación que presentarán la información de dicho producto.

Obra Preexistente: recurso audiovisual que es creado antes de ser concebido el producto en el cual será utilizado el mismo, para presentar la información que contendrá dicho producto.

Diseño: conjunto de elementos combinados que caracterizan y distinguen a los productos. Pueden ser distintivos de una colección de productos o de un producto en específico.

Marca: es un signo distintivo de cada producto que certifica su autenticidad en términos legales.

Ficha Técnica: es el documento que recoge los datos técnicos del producto, como son los nombres de los participantes, las herramientas utilizadas, breve descripción, entre otros.

Suplementos del Contrato: son documentos legales que agregan cláusulas nuevas al contrato previamente establecido, definiendo nuevas responsabilidades y compromisos entre las partes que antes no se habían establecido.

Anexos del Contrato: son documentos como el cronograma de ejecución, guión de contenido del software a desarrollar, o cualquier otro documento que explique y muestre algo de lo pactado dentro del Contrato.

Registro: acciones que se llevan a cabo para registrar los productos desarrollados, así como los diseños y las marcas empleadas en los mismos.

Solicitud de Servicio: se realiza una solicitud de servicio cuando es necesario desarrollar alguna obra audiovisual u otro componente destinado al producto, y no se cuente con el personal especializado en nuestro centro.

Licencia de Uso: se solicita una Licencia de Uso cuando es necesario utilizar en algún producto una obra audiovisual u otro componente desarrollado con anterioridad a la construcción del mismo. Con el objetivo de no evadir el derecho de autor.

Manual de Usuario: documento con un formato ajustado a su objetivo, está compuesto por: portada, Índice de contenidos, introducción, requerimientos básicos, descripción de las operaciones o funcionalidades, estructura general del sistema o diagrama de flujo y las palabras o términos en todos los casos asociadas al producto.

2.5.2 Diagrama de Dominio

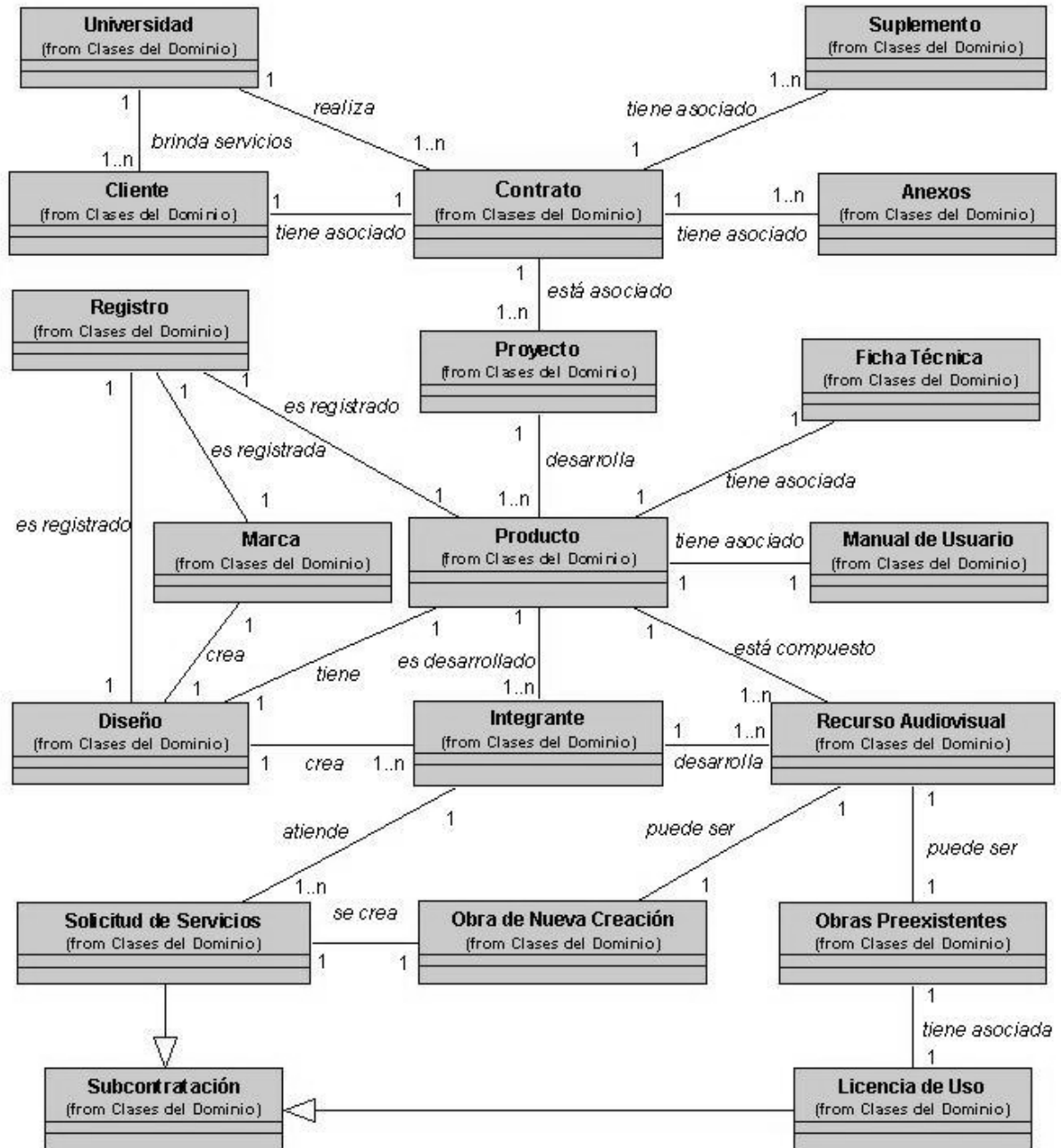


Figura 2.3 Diagrama de Dominio

2.6 Requerimientos funcionales.

Para poder lograr que el sistema a desarrollar sea realmente útil en las tareas para las cuales está predestinado es necesario establecer las funcionalidades del sistema propuesto, para lo cual deberá ser capaz de:

R1 Gestionar Cliente.

- 1.1 Insertar los datos del cliente.
- 1.2 Buscar clientes por determinados criterios.
- 1.3 Mostrar los datos del cliente.
- 1.4 Modificar los datos del cliente.
- 1.5 Eliminar cliente.

R2 Gestionar Proyecto.

- 2.1 Insertar los datos del proyecto.
- 2.2 Subir proyecto técnico.
- 2.3 Subir cronograma de ejecución.
- 2.4 Buscar proyectos por determinados criterios.
- 2.5 Mostrar los datos del proyecto.
- 2.6 Modificar los datos del proyecto.
- 2.7 Eliminar proyecto.
- 2.8 Eliminar proyecto técnico.
- 2.9 Eliminar cronograma de ejecución.

R3 Gestionar Producto.

- 3.1 Insertar los datos del producto.
- 3.2 Buscar productos por determinados criterios.
- 3.3 Mostrar los datos del producto.
- 3.4 Modificar los datos del producto.
- 3.5 Eliminar producto.

R4 Gestionar Ficha Técnica del producto.

- 4.1 Insertar datos de la ficha técnica.
- 4.2 Mostrar datos de la ficha técnica.
- 4.3 Modificar datos de la ficha técnica.
- 4.4 Generar el documento Ficha Técnica.
- 4.5 Eliminar ficha técnica.

R5 Gestionar Integrante.

- 5.1 Insertar los datos del integrante.
- 5.2 Buscar Integrantes por determinados criterios.
- 5.3 Mostrar los datos del integrante.
- 5.4 Modificar los datos del integrante.
- 5.5 Eliminar Integrante.

R6 Gestionar Contrato.

- 6.1 Insertar los datos del contrato.
- 6.2 Buscar contratos por determinados criterios.
- 6.3 Mostrar los datos del contrato.
- 6.4 Modificar los datos del contrato.
- 6.5 Eliminar contrato.
- 6.6 Subir suplementos del contrato.
- 6.7 Eliminar suplementos del contrato.
- 6.8 Subir anexos del contrato.
- 6.9 Eliminar anexos del contrato.

R7 Gestionar solicitud de servicio asociada a un proyecto.

- 7.1 Insertar datos de la solicitud de servicio.
- 7.2 Buscar solicitud de servicio por determinados criterios.
- 7.3 Mostrar los datos de la solicitud de servicio.

7.4 Modificar datos de la solicitud de servicio.

7.5 Eliminar solicitud de servicio.

7.6 Subir proforma de solicitud de servicio.

7.7 Eliminar proforma de solicitud de servicio.

R8 Gestionar licencia de uso asociada a un producto.

8.1 Insertar datos de la licencia de uso.

8.2 Buscar licencias de uso por determinados criterios

8.3 Mostrar los datos de la licencia de uso.

8.4 Modificar los datos de la licencia de uso

8.5 Eliminar licencia de uso.

8.6 Subir documento de licencia de uso.

8.7 Eliminar documento licencia de uso.

R9 Gestionar obra preexistente asociada a un producto.

9.1 Insertar datos de la obra preexistente.

9.2 Buscar obras preexistentes por determinados criterios

9.3 Mostrar los datos de la obra preexistente.

9.4 Modificar los datos de la obra preexistente.

9.5 Eliminar obra preexistente.

R10 Gestionar obra de nueva creación asociada a un producto.

10.1 Insertar los datos de la obra de nueva creación.

10.2 Buscar obras de nueva creación por determinados criterios

10.3 Mostrar los datos de la obra de nueva creación.

10.4 Modificar los datos de la obra de nueva creación.

10.5 Eliminar obra de nueva creación.

10.6 Inscribir obra de nueva creación como registrada en el CENDA.

R11 Gestionar diseño de productos o colecciones de productos.

- 11.1 Insertar los datos del diseño.
- 11.2 Buscar diseños por determinados criterios
- 11.3 Mostrar los datos del diseño.
- 11.4 Modificar los datos del diseño.
- 11.5 Eliminar diseño.
- 11.6 Inscribir el diseño como registrado en el CENDA.

R12 Gestionar marca de productos o colecciones de productos.

- 12.1 Insertar los datos de la marca.
- 12.2 Buscar marcas por determinados criterios
- 12.3 Mostrar los datos de la marca.
- 12.4 Modificar los datos de la marca.
- 12.5 Eliminar marca.
- 12.6 Subir copia de la marca.
- 12.7 Eliminar copia de la marca.
- 12.8 Inscribir la marca como registrada en el CENDA.

R13 Gestionar registro de software.

- 13.1 Emitir informe para el registro del software.
- 13.2 Inscribir el producto como registrado en el CENDA.

R14 Gestionar reportes.

- 14.1 Cantidad de productos (software) registrados.
- 14.2 Productos no registrados.
- 14.3 Cantidad de productos total.
- 14.4 Productos por proyecto.
- 14.5 Productos por proyectos y por clientes.

14.6 Obras audiovisuales de un producto seleccionado de acuerdo a su clasificación (preexistente y nueva creación).

R15 Administrar usuario.

15.1 Inscribir usuario.

15.2 Buscar usuario por determinados criterios.

15.3 Mostrar datos de usuario.

15.4 Modificar datos de usuario.

15.5 Eliminar usuario.

15.6 Restringir el acceso a las funcionalidades de acuerdo a los tipos de usuarios.

R16 Autenticar Usuario.

16.1 Permitir la autenticación de usuarios.

2.7 Requerimientos no funcionales.

RNF 1 Apariencia:

1.1 Diseño sencillo, de manera que se pueda utilizar el sistema sin mucho entrenamiento.

1.2 Diseño para la resolución 800x600, aunque preparado para verse en otras resoluciones.

RNF 2 Usabilidad:

2.1 El sistema podrá ser usado por cualquier persona, que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

2.2 Instalar el sistema, trae consigo una mayor rapidez de trabajo y por consiguiente un ahorro de materiales de oficina y personal.

RNF 3 Soporte:

3.1 Se requiere un servidor de bases de datos con las siguientes características:

3.1.1 Soporte para grandes volúmenes de datos y velocidad de procesamiento.

3.1.2 Tiempo de respuesta rápido en accesos concurrentes.

3.2 Versión de PHP 5.0 o superior.

3.3 Por parte del cliente se requiere un navegador capaz de interpretar Java Script.

RNF 4 Seguridad:

4.1 Establecer permisos para el acceso garantizando que solo acceda a la información quien tenga permiso.

4.2 Garantizar que las funcionalidades del sistema se muestren de acuerdo a los permisos del usuario que esté logueado.

4.3 Garantizar la protección ante acciones no autorizadas.

4.4 Verificación sobre acciones irreversibles (eliminación, modificación).

RNF 5 Confiabilidad:

5.1 Garantizar un tratamiento adecuado de excepciones y validaciones de las entradas de los usuarios.

5.2 Garantizar la recuperación ante fallos y errores.

RNF 6 Portabilidad:

6.1 Independencia de la plataforma.

2.8 Modelo de casos de uso del sistema.

Los casos de uso son documentos narrativos que se enfocan en mostrar una secuencia de acciones que desarrolla el actor en interacción con el sistema.

En la Figura 2.4 se muestra el diagrama de casos de uso del sistema, identificándose en el mismo tres actores del sistema y 16 casos de uso con los que interaccionan.

2.8.1 Diagrama de Casos de Uso del Sistema.

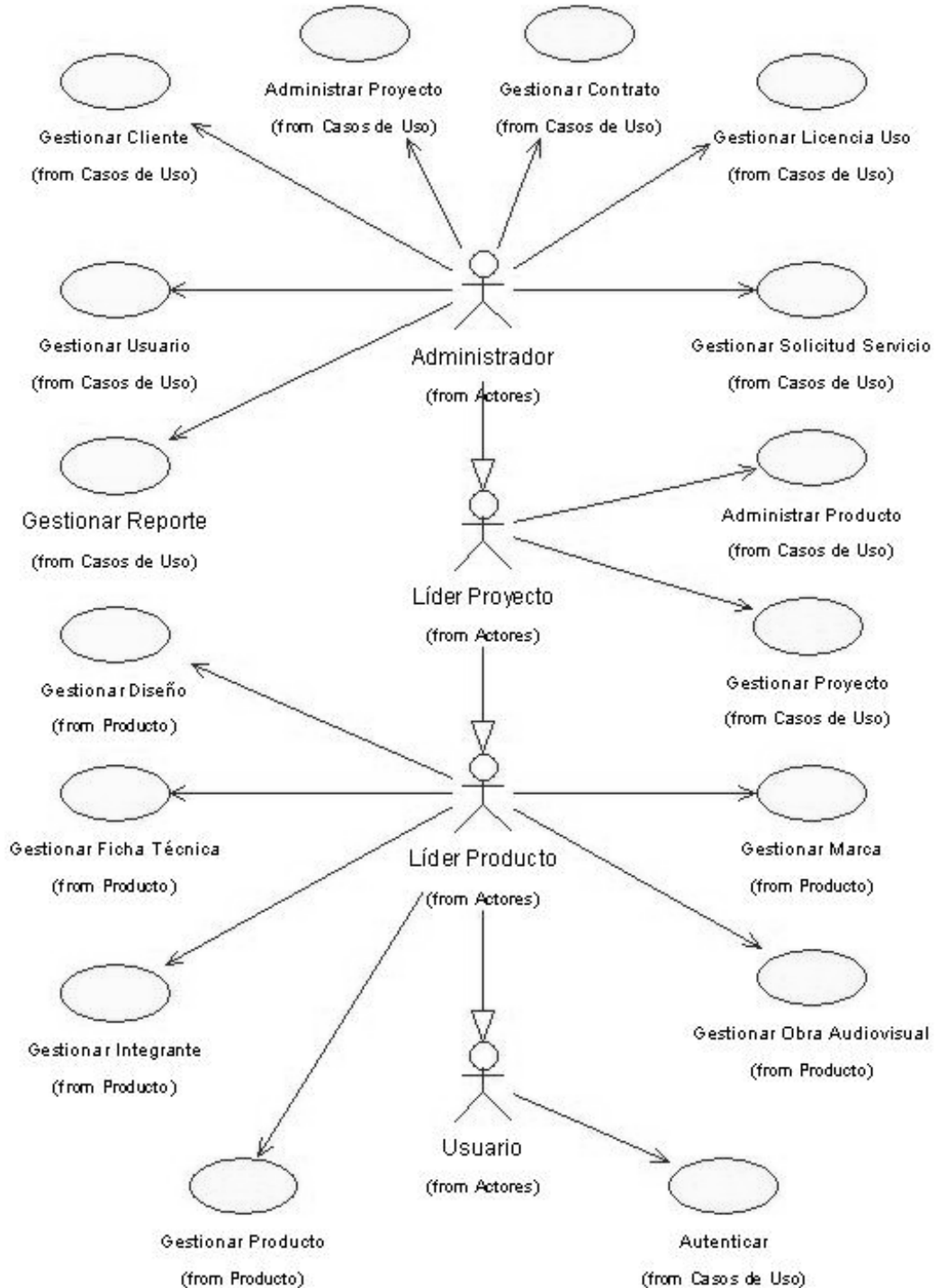


Figura 2.4 Diagrama de Casos de Uso del Sistema.

2.8.2 Actores del Sistema.

Actores	Justificación
Usuario	Cualquier persona que interactúe con el sistema.
Líder Producto	Usuario que está al frente de un producto y puede realizar acciones de modificación y consulta solo sobre su producto.
Líder Proyecto	Usuario que está al frente de un proyecto y puede realizar acciones de modificación y consulta solo sobre su proyecto y los productos de dicho proyecto.
Administrador	Usuario que puede realizar cualquier acción que realice el sistema.

2.8.3 Descripción de Casos de Uso del Sistema.

2.8.3.1 Autenticar Usuario.

Caso de Uso		Autenticar Usuario.
Actores	Usuario	
Propósito	El caso de uso se inicia cuando el usuario envía al sistema sus datos (nombre de usuario y contraseña). Se le informa si los datos fueron correctos o no y termina el caso de uso.	
Resumen	Garantizar que en todo momento se conozca quién está accediendo al sistema.	
Referencias	R16	
Precondiciones	El usuario ha introducido sus datos y elegido la opción de autenticarse.	
Poscondiciones	Se ha autenticado un usuario en el sistema.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El usuario se conecta al sistema.	1.1 Muestra el formulario necesario para que el usuario introduzca su usuario y contraseña.	
2. Introduce los datos requeridos y acepta la operación.	2.1 El sistema valida los datos introducidos.	
	2.2 Si los datos son correctos, le informa al usuario que ya está autenticado.	
Curso alternativo de los eventos.		
Acción 2.2	Si los datos no son válidos, muestra un mensaje de error e informa retornar a la acción 2.	

2.8.3.2 Gestionar Cliente.

Caso de Uso		Gestionar Cliente.
Actores	Administrador.	
Resumen	El CUS lo inicia el Administrador cuando desea gestionar la información de un cliente, selecciona la acción a realizar sobre el mismo, el sistema la ejecuta y termina el CUS.	

Propósito	Introducir los datos de un cliente, realizar búsquedas de clientes, para modificar, eliminar o consultar los datos de los mismos.	
Referencias	R1	
Precondiciones	El Administrador debe estar logueado.	
Poscondiciones	Cliente insertado en la Base de Datos. Datos del Cliente modificados en la Base de Datos. Datos del Cliente mostrados al Administrador. Datos del Cliente eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. Cuando el Administrador desea gestionar un cliente selecciona la opción "Cliente" ubicada en el Menú Principal.	1.1 El sistema muestra las opciones: Adicionar Cliente y Buscar Cliente.	
Escenario 1: Adicionar Cliente		
2. El Administrador selecciona la opción Adicionar Cliente.	2.1 El sistema muestra un formulario listo para introducir los datos del Cliente.	
3. El Administrador introduce los datos del Cliente y presiona el botón "Adicionar Cliente".	3.1 El sistema verifica si los datos introducidos por el Administrador son válidos.	
	3.2 Si los datos son válidos, el sistema inserta los datos en la BD y termina el CUS.	
Curso alterno de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al Administrador retornar a la acción 3.	
Escenario 2: Buscar Cliente.		
4. El Administrador selecciona la opción Buscar Cliente.	5.1 El sistema muestra un formulario con los parámetros de búsqueda por los cuales se puede localizar un Cliente.	
5. El Administrador selecciona el criterio de búsqueda (Nombre, Tipo, Siglas, Todos) y presiona el botón "Buscar Cliente"	5.1 El sistema verifica si existe algún Cliente de acuerdo al criterio de búsqueda seleccionado.	
	5.2 Si existen clientes de acuerdo al criterio de búsqueda seleccionado por el Administrador, el sistema lo muestra o los muestra en caso de ser varios, y termina el CUS.	
Curso alterno de los eventos.		
Acción 5.2	Si no existe ningún cliente de acuerdo con el criterio de búsqueda, el sistema muestra un mensaje informándole al Administrador retornar a la acción 5.	
Escenario 3: Consultar Datos.		
6. El administrador selecciona la opción "Buscar Cliente" luego realiza la acción 5, escoge el cliente deseado y presiona el botón "Mostrar Datos".	6.1 El sistema consulta la BD y muestra en un formulario con los datos del Cliente seleccionado.	

Escenario 4: Modificar Datos.	
7. El Administrador selecciona la opción "Buscar Cliente", luego realiza la acción 5, escoge el cliente deseado y presiona el botón "Modificar Datos".	7.1 El sistema muestra un formulario con los datos del cliente listos para ser modificados.
8. El Administrador hace los cambios pertinentes en los datos del cliente y presiona el botón "Actualizar Datos".	8.1 El sistema verifica que los datos modificados por el Administrador sean válidos.
	8.1 Si los datos son válidos, el sistema actualiza los datos en la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 9.1	Si los datos no son válidos el sistema muestra un mensaje de error, mostrando donde se encuentra el mismo, e informa al Administrador retornar a la acción 8.
Escenario 5: Eliminar Cliente.	
9. El Administrador selecciona la opción "Buscar Cliente", luego realiza la acción 5, escoge el cliente deseado y presiona el botón "Eliminar Cliente".	9.1 El sistema muestra un mensaje de advertencia para la acción a realizar, preguntando si desea o no eliminar dicho cliente.
10. El Administrador decide si desea o no eliminar el cliente.	10.1 Si el Administrador acepta, el sistema elimina el Cliente de la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 11.1	Si el Administrador decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.
Prioridad	Crítico

2.8.3.3 Administrar Proyecto.

Caso de Uso		Administrar Proyecto.
Actores	Administrador.	
Resumen	El CUS lo inicia el Administrador cuando desea gestionar un proyecto, selecciona la opción "Proyecto" escoge la acción a realizar, el sistema realiza las acciones correspondientes y termina el CUS.	
Propósito	Introducir los datos de un Proyecto, realizar búsquedas de Proyectos, para modificar, consultar o eliminar sus datos.	
Referencias	R2	
Precondiciones	El Usuario debe estar logueado.	
Poscondiciones	Proyecto insertado en la Base de Datos. Proyecto eliminado de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. Cuando el Administrador desea Administrar un proyecto, selecciona la opción "Proyecto" en el Menú Principal.	1.1 El sistema muestra las opciones: Adicionar Proyecto y Buscar Proyecto.	
Escenario 1: Adicionar Proyecto.		
2. Cuando el Administrador desea adicionar un proyecto, selecciona la opción "Adicionar	2.1 El sistema muestra un formulario listo para introducir los datos del Proyecto.	

Proyecto”.	
3. El Administrador introduce los datos del Proyecto, adjunta los documentos Proyecto Técnico y el Cronograma de Ejecución en caso de tenerlos en ese momento, y presiona el botón “Adicionar Proyecto”.	3.1 El sistema verifica que los datos introducidos sean válidos y verifica si los documentos adjuntos cumplen con los requisitos adecuados para colocarlos en la carpeta correspondiente a ese Proyecto.
	3.2 Si los datos son válidos, el sistema inserta los datos en la BD, coloca los documentos en la carpeta correspondiente y termina el CUS.
Curso alternativo de los eventos.	
Acción 3.2	Si los datos no son correctos, el sistema muestra un mensaje de error mostrando las causas de dicho error e informa al Administrador retornar a la acción 3.
Escenario 2: Buscar Proyecto.	
4. Cuando el Administrador desea buscar un proyecto, selecciona la opción “Buscar Proyecto” ubicada en el menú.	4.1 El sistema muestra un formulario con los parámetros de búsqueda necesarios para realizar dicha búsqueda.
5. El Administrador selecciona el criterio de búsqueda (Nombre, Tipo, Nombre del Cliente asociado, siglas del Cliente asociado, número de contrato asociado, Todos) y presiona el botón “Buscar Proyecto”	5.1 El sistema verifica si existe algún Proyecto de acuerdo al criterio de búsqueda seleccionado.
	5.2 Si existen Proyectos de acuerdo al criterio de búsqueda seleccionado, el sistema lo muestra, o los muestra en caso de ser varios y termina el CUS.
Curso alternativo de los eventos.	
Acción 5.2	Si no existe ningún Proyecto de acuerdo con el criterio de búsqueda seleccionado, el sistema muestra un mensaje informando la no existencia del mismo e informa al Administrador retornar a la acción 5.
Escenario 3: Eliminar Proyecto.	
6. El Administrador selecciona la opción “Buscar Proyecto”, luego realiza la acción 5, escoge el proyecto deseado y presiona el botón “Eliminar Proyecto”.	6.1 El sistema muestra un mensaje de advertencia para la acción a realizar, preguntando al Administrador si está seguro o no de realizar dicha acción.
7. El Administrador decide si desea o no eliminar el Proyecto.	7.1 Si el Administrador acepta, el sistema elimina el Proyecto de la BD, elimina la carpeta para documentos adjuntos y los documentos correspondientes al proyecto y termina el CUS.
Curso alternativo de los eventos.	
Acción 10.1	Si el Administrador decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.
Prioridad	Crítico

2.8.3.4 Gestionar Proyecto.

Caso de Uso		Gestionar Proyecto.
Actores	Administrador, Líder de Proyecto.	
Resumen	El CUS lo inicia el Administrador o el Líder de Proyecto cuando selecciona la opción "Proyecto", escoge la acción a realizar, el sistema la ejecuta y termina el CUS.	
Propósito	Realizar búsquedas de Proyectos, para modificar o consultar sus datos.	
Referencias	R2	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Datos del Proyecto modificados en la Base de Datos. Datos del Proyecto mostrados al usuario.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El usuario logueado selecciona la opción "Proyecto", ubicada en el Menú Principal.	1.2 El sistema muestra la opción: Buscar Proyecto.	
Escenario 1: Modificar Datos.		
2. Una vez realizada la acción 5, perteneciente al CUS Administrar Proyecto, Escenario 2, el usuario logueado selecciona el proyecto y presiona el botón "Modificar Datos",	2.1 El sistema muestra un formulario listo para modificar los datos del proyecto.	
3. El usuario logueado modifica los datos pertinentes, incluidos los documentos adjuntos.	3.1 El sistema verifica si los datos modificados son válidos.	
	3.2 Si los datos son válidos, el sistema modifica los datos y termina el CUS.	
Curso alternativo de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrándole al usuario logueado donde está dicho error e informa retornar a la acción 3.	
Escenario 2: Consultar Datos.		
4. Una vez realizada la acción 5, perteneciente al CUS Administrar Proyecto, Escenario 2, el usuario logueado selecciona el proyecto y presiona el botón "Consultar Datos".	4.1 El sistema muestra un formulario con los datos del Proyecto seleccionado y termina el CUS.	
Prioridad	Crítico	

2.8.3.5 Administrar Producto.

Caso de Uso		Administrar Producto.
Actores	Administrador, Líder de Proyecto.	
Resumen	El CUS lo inicia el Administrador o el Líder de Proyecto cuando selecciona la opción "Producto", escoge la acción a realizar sobre dicho producto, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de un Producto y realizar búsquedas de Productos para eliminar sus datos.	

Referencias	R3
Precondiciones	El usuario debe estar logueado.
Poscondiciones	Producto insertado en la Base de Datos. Producto eliminado de la Base de Datos.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1. El usuario logueado selecciona la opción "Producto".	1.1 El sistema muestra las opciones: Insertar Producto y Buscar Producto.
Escenario 1: Adicionar Producto.	
2. El usuario logueado selecciona la opción "Adicionar Producto".	2.1 El sistema muestra un formulario listo para insertar los datos del Producto.
3. El usuario logueado introduce los datos del Producto y presiona el botón "Adicionar Producto".	3.1 El sistema verifica que los datos introducidos sean válidos.
	3.2 Si los datos son válidos, el sistema inserta los datos en la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al usuario logueado retornar a la acción 3.
Escenario 2: Buscar Producto.	
4. El usuario logueado selecciona la opción "Buscar Producto".	4.1 El sistema muestra un formulario con los parámetros correspondientes, listo para realizar la búsqueda.
5. El usuario logueado selecciona el criterio de búsqueda (Nombre, Tipo, Nombre del Proyecto asociado, Tipo de Proyecto asociado, Número de Contrato asociado, Todos) y presiona el botón "Buscar Producto".	5.1 El sistema verifica si existe algún Producto de acuerdo al criterio de búsqueda seleccionado.
	5.2 Si existen Productos de acuerdo al criterio de búsqueda seleccionado, el sistema lo muestra o los muestra en caso de ser varios y termina el CUS.
Curso alternativo de los eventos.	
Acción 5.2	Si no existe ningún Producto de acuerdo con el criterio de búsqueda seleccionado, el sistema muestra un mensaje informándole al usuario logueado la inexistencia del mismo y retornar a la acción 5.
Escenario 3: Eliminar Producto.	
6. Una vez realizada la acción 5, el usuario logueado selecciona el producto y presiona el botón "Eliminar Producto".	6.1 El sistema muestra un mensaje de advertencia para la acción a realizar, preguntando si está seguro de la misma.
7. El usuario logueado decide si desea o no eliminar el Producto.	7.1 Si el usuario logueado acepta, el sistema elimina el Producto de la BD y termina el CUS.
Curso alternativo de los eventos.	

Acción 10.1	Si el usuario logueado decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.
Prioridad	Crítico

2.8.3.6 Gestionar Producto.

Caso de Uso		Gestionar Producto.
Actores	Administrador, Líder de Proyecto, Líder de Producto.	
Resumen	El CUS lo inicia el Administrador, Líder de Proyecto o el Líder de Producto cuando selecciona la opción "Producto", escoge la acción a realizar sobre dicho producto, el sistema la ejecuta y termina el CUS.	
Propósito	Realizar búsquedas de Productos, para modificar o consultar sus datos.	
Referencias	R3	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Datos del Producto modificados en la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El usuario logueado selecciona la opción "Producto", ubicada en el Menú Principal.	1.1 El sistema muestra la opción: Buscar Productos.	
Escenario 1: Modificar Datos.		
2. Una vez realizada la acción 5 del CUS Administrar Producto, Escenario 2, el usuario logueado selecciona el producto y presiona el botón "Modificar Datos".	2.1 El sistema muestra un formulario con los datos del producto listos para ser modificados.	
3. El usuario logueado modifica los datos pertinentes y presiona el botón "Actualizar Datos".	3.1 El sistema verifica si los datos modificados son válidos.	
	3.2 Si los datos son válidos, el sistema los modifica en la BD y termina el CUS.	
Curso alterno de los eventos.		
Acción 5.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrándole al usuario logueado donde está dicho error e informa retornar a la acción 5 del CUS Administrar Producto, Escenario 2.	
Escenario 2: Consultar Datos.		
4. Una vez realizada la acción 5 del CUS Administrar Producto, Escenario 2, el usuario logueado selecciona el producto deseado y presiona el botón "Consultar Datos",	4.1 El sistema muestra los datos del Producto seleccionado y termina el CUS.	
Prioridad	Crítico	

2.8.3.7 Gestionar Ficha Técnica.

Caso de Uso		Gestionar Ficha Técnica.
Actores	Administrador, Líder de Proyecto, Líder de Producto.	
Resumen	El CUS lo inicia el Administrador, el Líder de Proyecto o el Líder de Producto cuando selecciona la opción Gestionar Ficha Técnica, escoge la acción a realizar, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de la Ficha Técnica, realizar búsquedas de Fichas Técnicas para modificar datos, consultar datos o eliminar Fichas Técnicas.	
Referencias	R4	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	La Ficha Técnica insertada en la Base de Datos. Datos de la Ficha Técnica modificados en la Base de Datos. Datos de la Ficha Técnica mostrados al Líder de Producto. Datos de la Ficha Técnica eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. Cuando el usuario logueado desea gestionar una ficha técnica, selecciona la opción "Producto".	1.1 El sistema muestra la opción: Ficha Técnica.	
2. El usuario logueado selecciona la opción Ficha Técnica.	2.1 El sistema muestra las opciones: Adicionar Ficha Técnica y Buscar Ficha Técnica.	
Escenario 1: Adicionar Ficha Técnica.		
3. El usuario logueado selecciona la opción "Adicionar Ficha Técnica".	3.1 El sistema muestra un formulario listo para buscar el producto al cual se le asociará dicha Ficha Técnica. Ver CUS Gestionar Producto, Escenario 1.	
4. El usuario logueado selecciona el producto al cual se le asociará la Ficha Técnica y presiona el botón "Gestionar Ficha Técnica".	4.1 El sistema muestra un formulario listo para introducir los datos de la Ficha Técnica.	
5. El usuario logueado introduce los datos de la Ficha Técnica y presiona el botón "Adicionar Ficha Técnica".	5.1 El sistema verifica que los datos introducidos por el usuario logueado sean válidos.	
	3.2 Si los datos son válidos, el sistema inserta los datos en la BD y termina el CUS.	
Curso alterno de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al usuario logueado retornar a la acción 5.	
Escenario 2: Consultar Datos.		
6. Una vez realizada la acción 4, el usuario logueado presiona el botón "Consultar Datos".	6.1 El sistema muestra los datos generales de la Ficha Técnica seleccionada y termina el CUS.	

Escenario 3: Modificar Datos.	
7. Una vez realizada la acción 4, el usuario logueado presiona el botón “Modificar Datos”.	7.1 El sistema muestra un formulario con los datos de la Ficha Técnica listos para ser modificados.
8. El usuario logueado hace los cambios pertinentes y presiona el botón “Actualizar Datos”.	8.1 El sistema verifica que los datos modificados sean válidos.
	8.2 Si los datos son válidos, el sistema los actualiza en la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 8.2	Si los datos no son válidos el sistema muestra un mensaje de error mostrando donde está el dato erróneo e informa al Líder de Producto retornar a la acción 10.
Escenario 4: Eliminar Ficha Técnica.	
9. Una vez realizada la acción 4, el usuario logueado presiona el botón “Eliminar Ficha Técnica”.	9.1 El sistema muestra un mensaje de advertencia para la acción a realizar, preguntando si está seguro o no de la acción a realizar.
10. El usuario logueado decide si desea o no eliminar la Ficha Técnica.	10.1 Si el usuario logueado acepta, el sistema elimina la Ficha Técnica de la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 10.1	Si el Líder de Producto decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.8 Gestionar Integrante.

Caso de Uso		Gestionar Integrante.
Actores	Administrador, Líder de Proyecto, Líder de Producto.	
Resumen	El CUS lo inicia el Administrador, Líder de Proyecto o el Líder de Producto cuando selecciona la opción “Integrante”, escoge la acción a realizar sobre dicho integrante, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de los integrantes de los productos, realizar búsquedas de integrantes para modificar datos, consultar datos o eliminar integrantes.	
Referencias	R5	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Integrante insertado en la Base de Datos. Datos del Integrante modificados en la Base de Datos. Datos del Integrante mostrados al Líder de Producto. Datos del Integrante eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	

1. El usuario logueado selecciona la opción "Integrante" ubicada en el menú principal.	1.1 El sistema muestra las opciones: Insertar Integrante y Buscar Integrante.
Escenario 1: Adicionar Integrante	
2. El usuario logueado selecciona la opción Insertar Integrante.	2.1 El sistema muestra un formulario listo para realizar la búsqueda del producto asociado a dicho integrante.
3. El usuario logueado selecciona el producto y presiona el botón Adicionar Integrante.	3.1 El sistema muestra un formulario listo para insertar los datos del Integrante
4. El usuario logueado introduce los datos del Integrante y presiona el botón "Adicionar Integrante".	4.1 El sistema verifica que los datos introducidos sean válidos.
	4.2 Si los datos son válidos, el sistema inserta los datos en la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al usuario logueado retornar a la acción 3.
Escenario 2: Buscar Integrante.	
5. El usuario logueado selecciona la opción Buscar Integrante.	5.1 El sistema muestra un formulario con los parámetros de búsqueda pertinentes, listo para realizar la búsqueda.
6. El usuario logueado selecciona el criterio de búsqueda (Nombre, Tipo, CI, todos) y presiona el botón "Buscar Integrante"	6.1 El sistema verifica si existe algún Integrante de acuerdo al criterio de búsqueda seleccionado.
	6.2 Si existen Integrantes de acuerdo al criterio de búsqueda seleccionado, el sistema los muestra o lo muestra y termina el CUS.
Curso alternativo de los eventos.	
Acción 5.2	Si no existe ningún Integrante de acuerdo con el criterio de búsqueda, el sistema muestra un mensaje informándole al usuario logueado la inexistencia del mismo y retornar a la acción 6.
Escenario 3: Consultar Datos.	
7. Una vez realizada la acción 6, el usuario logueado selecciona el Integrante y presiona el botón "Consultar Datos".	7.1 El sistema muestra los datos del Integrante seleccionado y termina el CUS.
Escenario 4: Modificar Integrante.	
8. Una vez realizada la acción 6, el usuario logueado selecciona el Integrante y presiona el botón "Modificar Datos".	8.1 El sistema muestra el formulario con los datos del Integrante listos para ser modificados.
9. El usuario logueado hace los cambios pertinentes y presiona el botón "Actualizar Datos".	9.1 El sistema verifica que los datos modificados sean válidos.
	9.2 Si los datos son válidos, el sistema actualiza los datos en la BD y termina el CUS.
Curso alternativo de los eventos.	

Acción 8.2	Si los datos no son correctos el sistema muestra un mensaje de error mostrando donde se encuentra el mismo e informa al usuario logueado retornar a la acción 8.
Escenario 5: Eliminar Integrante.	
10. Una vez realizada la acción 6, el usuario logueado selecciona el Integrante y presiona el botón "Eliminar Integrante".	10.1 El sistema muestra un mensaje de advertencia, preguntando al usuario logueado si está seguro de la acción a realizar.
11. El usuario logueado decide si desea o no eliminar el Integrante.	11.1 Si el usuario logueado acepta, el sistema elimina el Integrante de la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 10.1	Si el usuario logueado decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.9 Gestionar Contrato.

Caso de Uso		Gestionar Contrato.
Actores	Administrador.	
Resumen	El CUS lo inicia el Administrador cuando selecciona la opción "Contrato", escoge la acción a realizar sobre dicho contrato, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de un Contrato, realizar búsquedas de contratos, para modificar, eliminar o examinar los datos de los mismos.	
Referencias	R6	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Contrato insertado en la Base de Datos. Datos del Contrato modificados en la Base de Datos. Datos del Contrato mostrados al Administrador. Datos del Contrato eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción Contrato.	1.1 El sistema muestra las opciones: Insertar Contrato y Buscar Contrato.	
Escenario 1: Adicionar Contrato.		
2. El Administrador selecciona la acción Adicionar Contrato.	2.1 El sistema muestra un formulario listo para la insertar los datos del Contrato.	
3. El Administrador introduce los datos del Contrato y presiona el botón "Adicionar Contrato".	3.1 El sistema verifica que los datos insertados sean válidos.	
	3.2 Si los datos son válidos, el sistema inserta los datos en la BD y termina el CUS.	
Curso alternativo de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje indicando donde se encuentra el error e informa al Administrador retornar a la	

	acción 3.
Escenario 2: Buscar Contrato.	
4. El Administrador selecciona la opción Buscar Contrato.	4.1 El sistema muestra un formulario con los parámetros de búsqueda pertinentes, listo para realizar la búsqueda.
5. El Administrador selecciona el criterio de búsqueda (Número de Contrato, Nombre Cliente, Siglas del Cliente, Todos) y presiona el botón "Buscar Contrato"	5.1 El sistema verifica si existe algún Contrato de acuerdo al criterio de búsqueda seleccionado.
	5.2 Si existen Contratos de acuerdo al criterio de búsqueda seleccionado, el sistema los muestra o lo muestra y termina el CUS.
Curso alterno de los eventos.	
Acción 5.2	Si no existe ningún Contrato de acuerdo con el criterio de búsqueda, el sistema muestra un mensaje informándole al Administrador la inexistencia del mismo y retornar a la acción 5.
Escenario 3: Consultar Datos.	
6. Una vez realizada la acción 5, el administrador selecciona el Contrato y presiona el botón "Consultar Datos".	6.1 El sistema muestra los datos identificativos del Contrato seleccionado.
Escenario 4: Modificar Datos.	
7. Una vez realizada la acción 5, el Administrador selecciona el Contrato y presiona el botón "Modificar Datos".	7.1 El sistema muestra un formulario con los datos del Contrato listos para ser modificados.
8. El Administrador hace los cambios pertinentes y presiona el botón "Actualizar Datos".	8.1 El sistema verifica que los datos insertados sean válidos.
	8.1 Si los datos son correctos el sistema actualiza los datos en la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 8.1	Si los datos no son correctos el sistema muestra un mensaje de error, mostrando donde está el mismo e informa al Administrador retornar a la acción 8.
Escenario 5: Eliminar Contrato.	
9. Una vez realizada la acción 5, el Administrador selecciona el Contrato y presiona el botón "Eliminar Contrato".	9.1 El sistema muestra un mensaje de advertencia, preguntando al Administrador si está seguro de la acción a realizar.
10. El Administrador decide si desea o no eliminar el Contrato.	10.1 Si el Administrador acepta, el sistema elimina el Contrato de la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 10.1	Si el Administrador decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

Escenario 6: Adicionar Anexos.	
11. Una vez realizada la acción 5, el Administrador selecciona el Contrato y presiona el botón “Adicionar Anexos”.	11.1 El sistema muestra un formulario listo para adjuntar el documento anexo.
12. El Administrador adjunta el documento y presiona el botón Aceptar.	12.1 El sistema verifica si el documento tiene las características idóneas para ser ubicado en la carpeta correspondiente al Contrato seleccionado.
	12.2 Si el documento cumple con las características, el sistema lo ubica en la carpeta correspondiente y termina el CUS.
Curso alternativo de los eventos.	
Acción 13.2	Si el documento no cumple con las características para ser colocado en la carpeta correspondiente, el sistema muestra un mensaje de error informando las causas del mismo y retornar a la acción 13.
Escenario 7: Eliminar Anexos.	
13. Una vez realizada la acción 5, el Administrador selecciona el Contrato y presiona el botón “Ver Anexos”	13.1 El sistema muestra un formulario con los documentos anexos correspondientes al contrato seleccionado.
14. El Administrador selecciona el anexo a eliminar y presiona el botón “Eliminar Anexo”.	14.1 El sistema muestra un mensaje de advertencia, preguntando si está seguro de la realización de dicha acción.
15. El Administrador decide si desea o no eliminar dicho anexo.	15.1 Si el Administrador acepta, el sistema elimina el anexo y Termina el CUS.
Curso alternativo de los eventos.	
Acción 17.1	Si el Administrador cancela, el sistema no realiza ninguna acción y termina el CUS.
Escenario 6: Adicionar Suplemento.	
16. Una vez realizada la acción 5, el Administrador selecciona el Contrato y presiona el botón “Adicionar Suplemento”.	16.1 El sistema muestra un formulario listo para adjuntar el documento suplementario.
17. El Administrador adjunta el documento y presiona el botón Aceptar.	17.1 El sistema verifica si el documento tiene las características idóneas para ser ubicado en la carpeta correspondiente al Contrato seleccionado.
	17.2 Si el documento cumple con las características, el sistema lo ubica en la carpeta correspondiente y termina el CUS.
Curso alternativo de los eventos.	
Acción 13.2	Si el documento no cumple con las características para ser colocado en la carpeta correspondiente, el sistema muestra un mensaje de error informando las causas del mismo y retornar a la acción 20.
Escenario 7: Eliminar Suplemento.	
18. Una vez realizada la acción 5, el Administrador selecciona el Contrato y	18.1 El sistema muestra un formulario con los documentos suplementarios correspondientes

presiona el botón “Ver Suplemento”.	al contrato seleccionado.
19. El Administrador selecciona el suplemento a eliminar y presiona el botón “Eliminar Suplemento”.	19.1 El sistema muestra un mensaje de advertencia, preguntando si está seguro de la realización de dicha acción.
20. El Administrador decide si desea o no eliminar dicho suplemento.	20.1 Si el Administrador acepta, el sistema elimina el suplemento y Termina el CUS.
Curso alterno de los eventos.	
Acción 24.1	Si el Administrador cancela, el sistema no realiza ninguna acción y termina el CUS.
Prioridad	Crítico

2.8.3.10 Gestionar Solicitud de Servicio.

Caso de Uso		Gestionar Solicitud de Servicio.
Actores	Administrador.	
Resumen	El CUS lo inicia el Administrador cuando selecciona la opción Solicitud de Servicio, escoge la acción a realizar sobre dicha Solicitud de Servicio, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de la Solicitud de Servicio, realizar búsquedas de Solicitudes de Servicio, para modificar datos, examinar los datos o eliminar las mismas.	
Referencias	R7	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Solicitud de Servicio insertada en la Base de Datos. Datos de la Solicitud de Servicio modificados en la Base de Datos. Datos de la Solicitud de Servicio mostrados al Administrador. Datos de la Solicitud de Servicio eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción “Solicitud de Servicio”.	1.1 El sistema muestra las opciones: Insertar Solicitud de Servicio y Buscar Solicitud de Servicio.	
Escenario 1: Adicionar Solicitud de Servicio.		
2. El Administrador selecciona la opción “Adicionar Solicitud de Servicio”.	2.1 El sistema muestra un formulario listo para buscar el contrato asociado a esa solicitud de servicio.	
3. El Administrador selecciona el contrato asociado y presiona el botón “Adicionar Solicitud de Servicio”	3.1 El sistema muestra un formulario listo para la adición de la Solicitud de Servicio.	
4. El Administrador introduce los datos de la Solicitud de Servicio, carga el documento Proforma de Solicitud de Servicio y presiona el botón “Adicionar Solicitud de Servicio”.	4.1 El sistema verifica que los datos introducidos sean válidos.	
	4.2 Si los datos son válidos, el sistema inserta los datos en la BD y termina el CUS.	
Curso alterno de los eventos.		

Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al Administrador retornar a la acción 4.
Escenario 2: Buscar Solicitud de Servicio.	
5. El Administrador selecciona la opción Buscar Solicitud de Servicio.	5.1 El sistema muestra un formulario con los parámetros de búsqueda correspondientes, listo para realizar la búsqueda.
6. El Administrador selecciona el criterio de búsqueda (Número de Solicitud, Número de Contrato, Nombre de la Entidad, Siglas de la Entidad, Todos) y presiona el botón "Buscar Solicitud de Servicio"	6.1 El sistema verifica si existe alguna Solicitud de Servicio de acuerdo al criterio de búsqueda seleccionado.
	6.2 Si existen Solicitudes de Servicio de acuerdo al criterio de búsqueda, el sistema las muestra o la muestra y termina el CUS.
Curso alternativo de los eventos.	
Acción 5.2	Si no existe ninguna Solicitud de Servicio de acuerdo con el criterio de búsqueda seleccionado, el sistema muestra un mensaje informándole al Administrador retornar a la acción 5.
Escenario 3: Consultar Datos.	
7. El administrador selecciona la Solicitud de Servicio y presiona el botón "Consultar Datos", una vez realizada la acción 5.	7.1 El sistema muestra los datos identificativos de la Solicitud de Servicio seleccionada y termina el caso de Uso.
Escenario 4: Modificar Datos.	
8 Una vez realizada la acción 6, el Administrador selecciona la Solicitud de Servicio deseada y presiona el botón "Modificar Datos",.	8.1 El sistema muestra un formulario con los datos de la Solicitud de Servicio listos para ser modificados.
9. El Administrador hace los cambios pertinentes y presiona el botón "Actualizar Datos".	9.1 El sistema verifica que los datos introducidos sean válidos.
	9.2 Si los datos son válidos el sistema actualiza los datos en la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 9.1	Si los datos no son válidos el sistema muestra un mensaje de error, mostrando donde se encuentra el mismo e informa al Administrador retornar a la acción 9.
Escenario 5: Eliminar Solicitud de Servicio.	
10. Una vez realizada la acción 6, el Administrador selecciona la Solicitud de Servicio deseada y presiona el botón "Eliminar Solicitud de Servicio".	10.1 El sistema muestra un mensaje de advertencia para la acción a realizar.
11. El Administrador decide si desea o no eliminar la Solicitud de Servicio.	11.1 Si el Administrador acepta, el sistema elimina la Solicitud de Servicio de la BD, elimina el documento Proforma de Solicitud

	de Servicio y termina el CUS.
Curso alternativo de los eventos.	
Acción 10.1	Si el Administrador decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.11 Gestionar Licencia de Uso.

Caso de Uso		Gestionar Licencia de Uso.
Actores	Administrador.	
Resumen	El CUS lo inicia el Administrador cuando selecciona la opción Licencia de Uso, escoge la acción a realizar sobre dicha Licencia de Uso, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de la Licencia de Uso, realizar búsquedas de Licencias de Uso, para modificar, consultar los datos de las mismas o eliminarlas.	
Referencias	R8	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Licencia de Uso insertada en la Base de Datos. Datos de la Licencia de Uso modificados en la Base de Datos. Datos de la Licencia de Uso mostrados al Administrador. Datos de la Licencia de Uso eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción "Licencia de Uso".	1.1 El sistema muestra las opciones: Insertar Licencia de Uso, Buscar Licencias de Uso, Consultar Datos, Modificar Datos y Eliminar Licencia de Uso.	
Escenario 1: Adicionar Licencia de Uso.		
2. El Administrador selecciona la opción "Adicionar Licencia de Uso".	2.1 El sistema muestra un formulario listo para buscar el contrato asociado.	
3. El Administrador selecciona el contrato asociado y presiona el botón "Adicionar Licencia de Uso"	3.1 El sistema muestra un formulario listo para insertar los datos de la Licencia de Uso.	
4. El Administrador introduce los datos de la Licencia de Uso, adjunta el documento Licencia de Uso y presiona el botón "Adicionar Licencia de Uso".	4.1 El sistema verifica que los datos introducidos sean válidos.	
	4.2 Si los datos son válidos, el sistema los inserta en la BD y termina el CUS.	
Curso alternativo de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al Administrador retornar a la acción 4.	
Escenario 2: Buscar Licencia de Uso.		

5. El Administrador selecciona la opción Buscar Licencia de Uso.	5.1 El sistema muestra un formulario con los parámetros de búsqueda pertinentes, listo para realizar la búsqueda.
6. El Administrador selecciona el criterio de búsqueda (Todos, proyecto, producto, otros) y presiona el botón "Buscar Licencia de Uso"	6.1 El sistema verifica si existe alguna Licencia de Uso de acuerdo al criterio de búsqueda seleccionado. 6.2 Si existen Licencias de Uso de acuerdo al criterio de búsqueda seleccionado, el sistema los muestra o lo muestra y termina el CUS.
Curso alterno de los eventos.	
Acción 5.2	Si no existe ninguna Licencia de Uso de acuerdo con el criterio de búsqueda seleccionado, el sistema muestra un mensaje informándole al Administrador retornar a la acción 6.
Escenario 3: Consultar Datos.	
7. El Administrador selecciona la Licencia de Uso deseada y presiona el botón "Consultar Datos", una vez realizada la acción 5.	7.1 El sistema muestra los datos identificativos de la Licencia de Uso seleccionada.
Escenario 4: Modificar Datos.	
8. El Administrador selecciona la Licencia de Uso deseada y presiona el botón "Modificar Datos", una vez realizada la acción 5.	8.1 El sistema muestra un formulario con los datos de la Licencia de Uso listos para ser modificados.
9. El Administrador hace los cambios pertinentes y presiona el botón "Actualizar Datos".	9.1 El sistema verifica que los datos modificados sean válidos. 9.2 Si los datos son válidos el sistema los actualiza en la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 9.1	Si los datos no son correctos el sistema muestra un mensaje de error mostrando donde se encuentra el mismo e informa al Administrador retornar a la acción 9.
Escenario 5: Eliminar Licencia de Uso.	
10. Una vez realizada la acción 6, el Administrador selecciona la Licencia de Uso deseada y presiona el botón "Eliminar Licencia de Uso".	10.1 El sistema muestra un mensaje de advertencia, preguntando si está seguro de la acción a realizar.
11. El Administrador decide si desea o no eliminar la Licencia de Uso.	11.1 Si el Administrador acepta, el sistema elimina la Licencia de Uso de la BD, elimina el documento Licencia de Uso y termina el CUS.
Curso alterno de los eventos.	
Acción 10.1	Si el Administrador decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.12 Gestionar Obra de Audiovisual.

Caso de Uso		Gestionar Obra Audiovisual.
Actores	Administrador, Líder de Proyecto, Líder de Producto.	
Resumen	El CUS inicia cuando el Administrador, Líder de Proyecto o el Líder de Producto selecciona la opción "Obra Audiovisual", escoge la acción a realizar sobre dicha Obra, el sistema realiza dicha acción y termina CUS.	
Propósito	Insertar los datos de las Obras Audiovisual, realizar búsquedas para modificar, consultar los datos o eliminar las obras.	
Referencias	R10	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Datos de las Obras Audiovisual insertados en la BD. Datos de las Obras Audiovisual modificados el la BD. Datos de las Obras Audiovisual mostrados. Datos de las Obras Audiovisual eliminados de la BD.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El usuario logueado selecciona la opción "Producto" ubicada en el Menú Principal.	1.1 El sistema muestra las opciones: Obra Audiovisual.	
2. El usuario logueado selecciona la opción "Obra Audiovisual".	2.1 El sistema muestra las opciones: Adicionar Obra Audiovisual y Buscar Obra Audiovisual.	
Escenario 1: Adicionar Obra Audiovisual.		
3. El usuario logueado selecciona la opción "Adicionar Obra Audiovisual".	3.1 El sistema muestra un formulario listo para buscar el producto asociado.	
4. El usuario logueado selecciona el producto asociado y presiona el botón "Adicionar Obra Audiovisual"	4.1 El sistema muestra un formulario listo para la adición de la Obra Audiovisual	
5. El usuario logueado introduce los datos de la Obra Audiovisual y presiona el botón "Adicionar Obra Audiovisual".	5.1 El sistema verifica que los datos sean válidos.	
	5.2 Si los datos son válidos, el sistema los inserta en la BD y termina el CUS.	
Curso alternativo de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al usuario logueado retornar a la acción 5.	
Escenario 2: Buscar Obra Audiovisual.		
6. El usuario logueado selecciona la opción "Buscar Obra Audiovisual".	6.1 El sistema muestra un formulario listo para realizar la búsqueda.	
7. El usuario logueado selecciona el criterio de búsqueda (Nombre, Tipo, Realizada Por, Estado, Todos) y presiona el botón "Buscar Obra Audiovisual"	7.1 El sistema verifica si existe alguna Obra Audiovisual de acuerdo al criterio de búsqueda seleccionado.	
	7.2 Si existen Obras Audiovisuales de acuerdo al criterio de búsqueda, el sistema las muestra o la muestra y termina el CUS.	

Curso alternativo de los eventos.	
Acción 5.2	Si no existe ninguna Obra Audiovisual de acuerdo con el criterio de búsqueda, el sistema muestra un mensaje informándole al usuario logueado la no existencia y retornar a la acción 6.
Escenario 3: Consultar Obra Audiovisual.	
8. Una vez realizada la acción 7, el usuario logueado selecciona la Obra Audiovisual y presiona el botón "Consultar Datos",.	8.1 El sistema muestra los datos de la Obra Audiovisual seleccionada.
Escenario 4: Modificar Obra Audiovisual.	
9. Una vez realizada la acción 7, el usuario logueado selecciona la Obra Audiovisual y presiona el botón "Modificar Datos",.	9.1 El sistema muestra el formulario con los datos de la Obra Audiovisual, listos para modificarlos.
10. El usuario logueado hace los cambios pertinentes y presiona el botón "Actualizar Datos".	10.1 El sistema verifica que los datos modificados sean válidos.
	10.2 Si los datos son válidos el sistema actualiza los datos en la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 8.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra el mismo e informa al usuario logueado retornar a la acción 10.
Escenario 5: Eliminar Obra Audiovisual.	
11. Una vez realizada la acción 7, el usuario logueado selecciona la Obra Audiovisual y presiona el botón "Eliminar Obra Audiovisual".	11.1 El sistema muestra un mensaje de advertencia preguntando si está seguro de la acción a realizar.
12. El usuario logueado decide si desea o no eliminar la Obra Audiovisual.	12.1 Si el usuario logueado acepta, el sistema elimina la Obra Audiovisual de la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 11.1	Si el usuario logueado decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.13 Gestionar diseño.

Caso de Uso	Gestionar Diseño.
Actores	Administrador, Líder de Proyecto, Líder de Producto.
Resumen	El CUS lo inicia el Administrador, Líder de Proyecto o el Líder de Producto cuando selecciona la opción Diseño, escoge la acción a realizar sobre los datos del diseño, el sistema la ejecuta y termina el CUS.
Propósito	Introducir los datos del Diseño, realizar búsquedas de Diseños para modificar, consultar o eliminar los datos del Diseño.
Referencias	R11
Precondiciones	El usuario debe estar logueado.

Poscondiciones	Diseño insertado en la Base de Datos. Datos del Diseño modificados en la Base de Datos. Datos del Diseño mostrados al Líder de Producto. Datos del Diseño eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El usuario logueado selecciona la opción "Diseño".	1.1 El sistema muestra las opciones: Insertar Diseño y Buscar Diseño.	
Escenario 1: Adicionar Diseño.		
2. El usuario logueado selecciona la acción "Adicionar Diseño".	2.1 El sistema muestra un formulario listo para buscar el producto asociado.	
3. El usuario logueado selecciona el producto asociado y presiona el botón "Adicionar Diseño"	3.1 El sistema muestra un formulario listo para insertar los datos del Diseño.	
4. El usuario logueado introduce los datos del Diseño y presiona el botón "Adicionar Diseño".	4.1 El sistema verifica que los datos insertados sean válidos.	
	3.2 Si los datos son correctos, el sistema los inserta en la BD y termina el CUS.	
Curso alterno de los eventos.		
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al usuario logueado retornar a la acción 4.	
Escenario 2: Buscar Diseño.		
5. El usuario logueado selecciona la opción Buscar Diseño.	5.1 El sistema muestra un formulario con los parámetros de búsqueda pertinentes, listo para realizar la búsqueda.	
6. El usuario logueado selecciona el criterio de búsqueda (Nombre, Registrar, Estado, Todos) y presiona el botón "Buscar Diseño"	6.1 El sistema verifica si existe algún Diseño de acuerdo al criterio de búsqueda seleccionado.	
	6.2 Si existen Diseños de acuerdo al criterio de búsqueda, el sistema los muestra o lo muestra y termina el CUS.	
Curso alterno de los eventos.		
Acción 5.2	Si no existe ningún Diseño de acuerdo con el criterio de búsqueda, el sistema muestra un mensaje informándole al usuario logueado la inexistencia y retornar a la acción 5.	
Escenario 3: Consultar Datos.		
7. Una vez realizada la acción 6, el usuario logueado selecciona el Diseño y presiona el botón "Consultar Datos".	7.1 El sistema muestra los datos del Diseño seleccionado.	
Escenario 4: Modificar Datos.		
8. Una vez realizada la acción 6, el usuario logueado selecciona el Diseño y presiona el botón "Modificar Datos".	8.1 El sistema muestra el formulario con los datos del Diseño listos para modificar.	

9. El usuario logueado hace los cambios pertinentes y presiona el botón "Actualizar Datos".	9.1 El sistema verifica que los datos sean válidos.
	9.2 Si los datos son válidos el sistema actualiza los datos en la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 8.2	Si los datos no son válidos el sistema muestra un mensaje de error e informa al usuario logueado retornar a la acción 9.
Escenario 5: Eliminar Diseño.	
10. Una vez realizada la acción 6, el usuario logueado selecciona el Diseño y presiona el botón "Eliminar Diseño",.	10.1 El sistema muestra un mensaje de advertencia preguntando al usuario logueado si desea o no realizar dicha acción.
11. El usuario logueado decide si desea o no eliminar el Diseño.	11.1 Si el usuario logueado acepta, el sistema elimina el Diseño de la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 10.1	Si el usuario logueado decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.14 Gestionar Marca.

Caso de Uso		Gestionar Marca.
Actores	Administrador, Líder de Proyecto, Líder de Producto.	
Resumen	El CUS lo inicia el Administrador, el Líder Proyecto o el Líder de Producto cuando selecciona la opción de Marca, escoge la acción a realizar sobre dicha Marca, el sistema la ejecuta y termina el CUS.	
Propósito	Introducir los datos de la Marca, realizar búsquedas de Marcas para modificar, consultar o eliminar los datos de la Marca.	
Referencias	R12	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Marca insertada en la Base de Datos. Datos de la Marca modificados en la Base de Datos. Datos de la Marca mostrados al Líder de Producto. Datos de la Marca eliminados de la Base de Datos.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El usuario logueado selecciona la opción Gestionar Marca.	1.1 El sistema muestra las opciones: Insertar Marca y Buscar Marca.	
Escenario 1: Adicionar Marca		
2. El usuario logueado selecciona la acción Adicionar Marca.	2.1 El sistema muestra un formulario listo para buscar el producto asociado.	
3. El usuario logueado selecciona el producto y presiona el botón "Adicionar Marca"	3.1 El sistema muestra un formulario listo para insertar los datos de la Marca.	
4. El usuario logueado introduce los datos	4.1 El sistema verifica que los datos	

de la Marca y presiona el botón “Adicionar Marca”.	introducidos sean válidos.
	3.2 Si los datos son válidos, el sistema los inserta en la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 3.2	Si los datos no son válidos, el sistema muestra un mensaje de error mostrando donde se encuentra dicho error e informa al usuario logueado retornar a la acción 4.
Escenario 2: Buscar Marca.	
5. El usuario logueado selecciona la opción Buscar Marca.	5.1 El sistema muestra un formulario con los parámetros correspondientes, listo para realizar la búsqueda.
6. El usuario logueado selecciona el criterio de búsqueda (Nombre, Registrar, Estado, Todos) y presiona el botón “Buscar Marca”	6.1 El sistema verifica si existe alguna Marca de acuerdo al criterio de búsqueda seleccionado.
	6.2 Si existen Marcas de acuerdo al criterio de búsqueda, el sistema los muestra o lo muestra y termina el CUS.
Curso alterno de los eventos.	
Acción 5.2	Si no existe ninguna Marca de acuerdo con el criterio de búsqueda, el sistema muestra un mensaje informándole al Líder de Producto la inexistencia de la misma y retornar a la acción 6.
Escenario 3: Consultar Datos.	
7. Una vez realizada la acción 6, el usuario logueado selecciona la Marca y presiona el botón “Consultar Datos”.	7.1 El sistema muestra los datos de la Marca seleccionado.
Escenario 4: Modificar Marca.	
8. Una vez realizada la acción 6, el usuario logueado selecciona la Marca y presiona el botón “Modificar Datos”.	8.1 El sistema muestra el formulario con los datos de la Marca listos para ser modificados.
9. El usuario logueado hace los cambios pertinentes y presiona el botón “Actualizar Datos”.	9.1 El sistema verifica que los datos modificados sean válidos.
	9.2 Si los datos son válidos el sistema los actualiza en la BD y termina el CUS.
Curso alterno de los eventos.	
Acción 8.2	Si los datos no son válidos el sistema muestra un mensaje de error mostrando donde se encuentra el mismo, e informa al usuario logueado retornar a la acción 9.
Escenario 5: Eliminar Marca.	
10. Una vez realizada la acción 6, el usuario logueado selecciona la Marca y presiona el	10.1 El sistema muestra un mensaje de advertencia para la acción a realizar.

botón "Eliminar Marca".	
11. El usuario logueado decide si desea o no eliminar la Marca.	11.1 Si el usuario logueado acepta, el sistema elimina la Marca de la BD y termina el CUS.
Curso alternativo de los eventos.	
Acción 10.1	Si el usuario logueado decide cancelar la acción, termina el CUS y el sistema no realiza ninguna acción.

2.8.3.15 Gestionar Reportes.

Caso de Uso		Gestionar Reportes.
Actores	Administrador	
Resumen	El caso de uso se inicia cuando el Administrador selecciona la opción de Reportes, escoge el tipo de reporte que desea consultar, el sistema muestra los datos solicitados por el Administrador y termina el CUS.	
Propósito	Mostrar al Administrador reportes que le ayudarán en la realización y correcto funcionamiento de la estrategia técnico legal.	
Referencias	R14	
Precondiciones	El Administrador debe estar logueado.	
Poscondiciones	Datos a consultar mostrados al Administrador.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción "Reportes".	1.1 El sistema muestra las Opciones siguientes: Cantidad de productos registrados, Productos no registrados, Cantidad de productos total, Productos por proyecto, Productos por proyectos y por clientes, Obras audiovisuales de un producto.	
Escenario 1: Cantidad de Productos Registrados.		
2. El Administrador selecciona la opción Cantidad de Productos Registrados, después de haber realizado acción 1.	2.1 El sistema verifica si existe algún producto registrado.	
	2.2 Si existen productos registrados el sistema muestra la información solicitada.	
Curso alternativo de los eventos.		
Acción 2.2	Si no existen, muestra un mensaje de error informando la no existencia de productos registrados.	
Escenario 2: Productos no Registrados.		
3. El Administrador selecciona la opción Productos no Registrados.	3.1 El sistema verifica si existen productos no registrados.	
	3.2 Si existen, el sistema muestra los productos no registrados.	
Curso alternativo de los eventos		
Acción 3.2	Si no existen productos que no estén registrados, el sistema muestra un mensaje informándole al cliente la no existencia de los	

	mismos.
Escenario 3: Cantidad de productos total.	
4. El Administrador selecciona la opción Cantidad de productos total.	4.1 El sistema muestra todos los Productos existentes en la BD.
Curso alternativo de los eventos.	
Acción 4.1	Si no existen productos en la BD, el sistema muestra un mensaje informándole al Administrador la no existencia de productos en la BD.
Escenario 4: Productos por Proyecto	
5. El Administrador selecciona la opción Productos por Proyecto.	5.1 El sistema muestra el Formulario listo para seleccionar el Proyecto del cual se necesita conocer los productos que contiene. Ver CUS Gestionar Proyecto, Escenario 1.
6. El Administrador selecciona el proyecto y presiona el botón "Buscar Productos".	6.1 El sistema verifica si el proyecto seleccionado contiene algún producto asociado.
	6.2 Si Existen productos asociados al proyecto seleccionado, el sistema los muestra.
Curso alternativo de los eventos	
Acción 6.2	Si no existen productos asociados al proyecto seleccionado, el sistema muestra un mensaje informando la situación y retorna al Administrador a la acción 6.
Escenario 5: Productos por Proyectos y por Clientes.	
7. El Administrador selecciona la opción Productos por Proyecto y Por Clientes.	7.1 El sistema muestra el formulario correspondiente, listo para seleccionar el proyecto y el cliente.
8. El Administrador selecciona el proyecto y el cliente deseado y presiona el botón "Buscar Productos".	8.1 El sistema verifica si existe algún producto asociado al proyecto y al cliente seleccionado.
	8.2 Si existe algún producto de acuerdo con la selección, el sistema los muestra.
Curso alternativo de los eventos	
Acción 8.2	Si no existen Productos de acuerdo con la selección el sistema muestra un mensaje informando la situación y retorna al Administrador a la acción 8.
Escenario 6: Obras Audiovisuales por Producto.	
9. El Administrador selecciona la opción Obras Audiovisuales por Producto.	9.1 El sistema muestra un formulario listo para la búsqueda del producto del cual se desea conocer sus Obras Audiovisuales. Ver CUS Gestionar Producto, Escenario 1.
10. El Administrador selecciona el producto deseado una vez encontrado, y presiona el botón "Buscar Obras Audiovisuales".	10.1 El sistema verifica si existen Obras audiovisuales asociadas al producto.
	10.2 Si existen Obras Audiovisuales asociadas al producto, el sistema las muestra.

Curso alterno de los eventos	
Acción 10.2	Si no existen Obras Audiovisuales asociadas al producto, el sistema muestra un mensaje informando la situación.

2.8.3.16 Administrar Usuario.

Caso de Uso		Administrar Usuario.
Actores	Administrador	
Resumen	El caso de uso inicia cuando el administrador selecciona la opción de administrar usuario, el sistema le muestra la interfaz correspondiente y las acciones que podrá realizar, luego el sistema ejecuta la acción seleccionada y termina el caso de uso.	
Propósito	Crear nuevos usuarios para que utilicen el sistema, especificando el rol que permitirá establecer niveles de permisos, realizar búsquedas de usuarios para modificar, eliminar o examinar los perfiles de los usuarios creados.	
Referencias	RF 15	
Precondiciones	El usuario debe estar logueado.	
Poscondiciones	Usuario insertado en la Base de Datos. Mostrar los datos del usuario seleccionado. Información del Usuario modificada en la Base de Datos. Usuario eliminado de la Base de Datos. Definido el nivel de permisos de los usuarios.	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El administrador selecciona la opción de Administrar Usuario.	1.1 El sistema muestra las opciones: Adicionar Usuario y Buscar Usuario.	
Escenario 1: Adicionar Usuario		
2. El administrador selecciona la opción de Adicionar Usuario.	2.1 El sistema muestra un formulario listo para insertar los datos del usuario.	
3. El Administrador introduce los datos del usuario.	3.1 El sistema verifica si los datos introducidos son válidos.	
	3.2 Si los datos son válidos, los inserta en la BD, y termina el Caso de Uso.	
Curso alterno de los eventos		
Acción 3.1	Si los datos no son correctos, el sistema muestra un mensaje de error mostrando donde está el error e indica al Administrador retornar a la acción 3.	
Escenario 2: Buscar Usuario		
4. El Administrador selecciona la opción de Buscar Usuario.	4.1 El sistema muestra un formulario con los parámetros correspondientes, listo para realizar la búsqueda.	
5. El Administrador selecciona el criterio de búsqueda (rol, usuario) y presiona el botón "Buscar Usuario".	5.1 El sistema verifica si existe algún usuario de acuerdo al criterio de búsqueda seleccionado.	

	5.2 Si existe un usuario de acuerdo con el criterio de búsqueda seleccionado, el sistema lo muestra y termina el CUS.
Curso alterno de los eventos	
Acción 5.2	Si no existe un usuario de acuerdo al criterio de búsqueda seleccionado el sistema muestra un mensaje informando la no existencia del mismo, y retorna al Administrador a la acción 5.
Escenario 3: Modificar Usuario	
6. Una vez realizada la acción 6, el administrador selecciona el usuario al cual desea modificar los datos y presiona el botón "Modificar Usuario".	6.1 El sistema localiza los datos del usuario seleccionado y los muestra, listos para ser modificados.
7. El Administrador realiza los cambios necesarios a los datos y presiona el botón: "Actualizar Datos".	7.1 El sistema verifica los datos modificados por el Administrador.
	7.2 Si los datos son válidos el sistema actualiza los datos del usuario en la Base de Datos (BD) y termina el CUS.
Curso alterno de los eventos	
Acción 7.2	Si los datos modificados por el Administrador no son válidos, el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al Administrador retornar a la acción 7.
Escenario 4: Mostrar Datos	
8. Una vez realizada la acción 6, el administrador selecciona el usuario al cual desea modificar los datos y presiona el botón "Mostrar Datos".	8.1 El sistema localiza los datos del usuario seleccionado, los muestra y termina el CUS.
Escenario 5: Eliminar Usuario.	
9. Una vez realizada la acción 5, el Administrador presiona el botón "Eliminar Usuario".	9.1 El sistema muestra un mensaje de advertencia, para la acción a realizar.
10. El Administrador confirma si desea o no eliminar el usuario o los usuarios.	10.1 Si el administrador acepta, el sistema elimina los usuarios seleccionados, de la Base de Datos, y termina el CUS.
Curso alterno de los eventos	
Acción 13.1	Si el Administrador cancela la acción, se culmina el CUS sin ejecutar ninguna acción.

2.9 Conclusiones del Capítulo.

En este capítulo se hizo una breve exposición de la situación actual en el ámbito de la legalidad del software educativo y se expusieron las razones por las cuales se pretende desarrollar el Sistema. Se realizó la propuesta de solución a través de un Modelo de Dominio, identificándose los conceptos involucrados en el proceso, se exponen además los requisitos funcionales y no funcionales de la aplicación que se pretende desarrollar y la modelación de la misma, a través de los casos de uso de sistema y su descripción. Lo cual constituye el punto de partida en la construcción del Sistema.

CAPÍTULO **3**: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

En el presente capítulo se diseñan los diagramas de clases del análisis, con el objetivo de brindar una visión preliminar y de fácil entendimiento de los procesos involucrados en el Sistema. Se realiza un modelo de diseño, donde se explica detalladamente el funcionamiento específico de los artefactos que intervienen en la construcción de la Aplicación Web. Se exponen los patrones de diseño a utilizar y como se lleva a cabo el tratamiento de errores, la seguridad, diseño de la interfaz y como está concebida la ayuda que brindará el Sistema.

3.2 Diagramas de Clases del Análisis.

Se diseñaron los diagramas de clases del análisis para crear el modelo conceptual inicial de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. Los mismos se encuentran ubicados en el Anexo 1 debido a su extensión y cantidad.

3.3 Modelo de Diseño.

3.3.1 Diagramas de Clases Web.

Se construyeron varios diagramas de clases Web por cada caso de uso del sistema, con el objetivo de brindar un mayor entendimiento acerca de cómo interaccionan los componentes inmersos en cada uno de ellos en su funcionamiento dentro del proceso. Los mismos se encuentran en el Anexo 2 debido a su extensión y cantidad.

3.3.2 Descripción de Clases del Diseño.

3.3.2.1 Descripción de Clases Entidad.

Nombre: ECliente	
Tipo de clase: Entidad	
Atributo	Tipo
Id_Cliente	Integer
Nombre	String
Siglas	String
Objeto_Social	String
Fax	String
Teléfono	String
Tipo_Cliente	String
Obj_Pais	Objeto País
Obj_Representante	Objeto Representante
Nombre	Descripción
Ecliente(\$aNombre, \$aSiglas, \$aObjetoSocial, \$aFax, \$aTelefono, \$aObj_Pais, \$aTipo_Cliente, \$aObj_Representante)	Constructor de la clase Cliente.
Get_Nombre()	Retorna el nombre del cliente.
Get_Siglas()	Retorna las siglas del cliente.
Get_Objeto_Social()	Retorna el Objeto Social.
Get_Fax()	Retorna el Fax del cliente.
Get_Telefono()	Retorna el teléfono del cliente.
Get_Obj_Pais()	Retorna el objeto País.
Get_Tipo_Cliente	Retorna el tipo de cliente.
Get_Id_Cliente()	Retorna el id del cliente.
Get_Obj_Representante()	Retorna el objeto Representante.
Set_Obj_Pais(\$aObj)	Modifica el objeto País.
Set_Obj_Representante(\$aObj)	Modifica el objeto Representante.
Set_Id_Cliente(\$ald)	Modifica el id del cliente.
Set_Nombre(\$aNomb)	Modifica el nombre del cliente.
Set_Siglas(\$aSiglas)	Modifica las siglas del cliente.
Set_Objeto_Social(\$aObjeto_Social)	Modifica el objeto social.
Set_Fax(\$aFax)	Modifica el fax del cliente.
Set_Telefono(\$aTelefono)	Modifica el cliente.
Set_Tipo_Cliente(\$aTipo_Cliente)	Modifica el tipo de cliente.

Nombre: Ecliente_Nacional	
Tipo de clase: Entidad	
Atributo	Tipo
Resolucion_Creacion	String
Obj_Cuenta_Bancaria_Cuc	String
Obj_Cuenta_Bancaria_Cup	String
Nombre	Descripción
Ecliente_Nacional(\$aTipo_Cliente, \$aResolucion_Creacion, \$aObj_Cuenta_Bancaria_Cuc, \$aObj_Cuenta_Bancaria_Cup, \$aNombre, \$aSiglas, \$aObjetoSocial, \$aFax, \$aTelefono, \$aObj_Pais, \$aObj_Representante)	Constructor de la clase Cliente Nacional.
Get_Resolucion_Creacion()	Retorna la resolución de creación.
Set_Resolucion_Creacion(\$aResolucion)	Modifica la resolución de creación.
Get_Cuenta_Bancaria_Cuc()	Retorna la cuenta bancaria en CUC.
Get_Cuenta_Bancaria_Cup()	Retorna la cuenta bancaria en CUP.
Set_Cuenta_Bancaria_Cuc(\$aCuenta)	Modifica la cuenta bancaria en CUC.
Set_Cuenta_Bancaria_Cup(\$aCuenta)	Modifica la cuenta bancaria en CUP.

Nombre: Ecliente_Extranjero	
Tipo de clase: Entidad	
Atributo	Tipo
Acta_Representacion	String
Numero_Registro	String
Obj_Cuenta_Bancaria	String
Nombre	Descripción
Ecliente_Extranjero(\$aTipo_Cliente, \$aActa_Representacion, \$aNumero_Registro, \$aObj_Cuenta_Bancaria, \$aNombre, \$aSiglas, \$aObjetoSocial, \$aFax, \$aTelefono, \$aObj_Pais, \$aObj_Representante)	Constructor de la clase Cliente Extranjero.
Get_Acta_Representacion()	Retorna el Acta de Representación.
Get_Numero_Registro()	Retorna el Número de Registro.
Get_Cuenta_Bancaria()	Retorna la cuenta bancaria.
Set_Cuenta_Bancaria(\$aCuenta_Bancaria)	Modifica la cuenta bancaria.
Set_Acta_Representacion(\$aActa_Representacion)	Modifica el Acta de Representación.
Set_Numero_Registro(\$aNum)	Modifica el número de registro.

Nombre: EProyecto	
Tipo de clase: Entidad	
Atributo	Tipo
Id_Proyecto	Integer
Obj_Cliente	Objeto
Obj_Contrato	Objeto
Tipo_Proyecto	String
Nombre_Proyecto	String
Proyecto_Tecnico	String
Cronograma_Ejecucion	String
Obj_Responsable_Cliente	Objeto
Obj_Responsable_Uci	Objeto
Nombre	Descripción
EProyecto(\$aTipo_Proyecto, \$aNombre_Proyecto, \$aProyecto_Tecnico , \$aCronograma_Ejecucion , \$aObj_Cliente , \$aObj_Responsable_Cliente , \$aObj_Responsable_Uci)	Constructor de la clase.
Get_Id_Proyecto()	Retorna el valor del atributo Id_Proyecto
Get_Obj_Cliente()	Retorna el objeto cliente.
Get_Obj_Contrato()	Retorna el objeto contrato.
Get_Tipo_Proyecto()	Retorna el valor del atributo Tipo_Proyecto.
Get_Nombre_Proyecto()	Retorna el valor del atributo Nombre_Proyecto
Get_Proyecto_Tecnico()	Retorna el valor del atributo Proyecto_Tecnico
Get_Cronograma_Ejecucion()	Retorna el valor del atributo Cronograma_Ejecucion
Get_Responsable_Cliente()	Retorna el valor del atributo Responsable_Cliente
Get_Responsable_Uci()	Retorna el valor del atributo Responsable_Uci
Set_Id_Proyecto(\$ald)	Modifica el valor del atributo Id_Proyecto
Set_Obj_Cliente(\$aObj)	Modifica el objeto cliente.
Set_Obj_Contrato(\$aObj)	Modifica el objeto contrato.
Set_Proyecto_Tecnico(\$aPT)	Modifica el valor del atributo Proyecto_Tecnico
Set_Cronograma_Ejecucion(\$aCE)	Modifica el valor del atributo Cronograma_Ejecucion
Set_Tipo_Proyecto(\$aTipo)	Modifica el valor del atributo Tipo_Proyecto
Set_Nombre_Proyecto(\$aNombre)	Modifica el valor del atributo Nombre_Proyecto



Set_Responsable_Cliente(\$aObj)	Modifica el objeto Responsable_Cliente
Set_Responsable_Uci(\$aObj)	Modifica el objeto Responsable_Uci

Nombre: EProducto	
Tipo de clase: Entidad	
Atributo	Tipo
Id_Producto	Integer
Nombre_Producto	String
Certificado_Calisoft	Integer
Obj_Archivo_Manual_Usuario	Objeto
Tiene_Ejemplar	Integer
Obj_Proyecto	Objeto
Obj_Disenio	Objeto
Obj_Marca	Objeto
Obj_Tipo_Producto	Objeto
Nombre	Descripción
EProducto(\$aObj_Proyecto, \$aNombre_Producto, \$aObj_Tipo_Producto, \$aCertificado_Calisoft, \$aObj_Archivo_Manual_Usuario, \$aEjemplar)	Constructor de la clase.
Get_Id_Producto()	Retorna el valor del atributo Id_Producto.
Get_Nombre_Producto()	Retorna el valor del atributo Nombre_Producto.
Get_Esta_Certificado()	Retorna el valor del atributo Esta_Certificado.
Get_Obj_Manual_Usuario()	Retorna el objeto Manual de Usuario.
Get_Tiene_Ejemplar()	Retorna el valor del atributo Tiene_Ejemplar.
Get_Obj_Disenio()	Retorna el objeto Diseno.
Get_Obj_Marca()	Retorna el objeto Marca.
Get_Obj_Tipo_Producto()	Retorna el objeto Tipo_Producto.
Get_Obj_Proyecto()	Retorna el objeto Proyecto.
Set_Id_Producto(\$ald)	Modifica el valor del atributo Id_Producto.
Set_Nombre_Producto(\$aNomb)	Modifica el valor del atributo Nombre_Producto.
Set_Esta_Calificado(\$Calificado)	Modifica el valor del atributo Esta_Calificado.
Set_Obj_Manual_Usuario(\$aObj_Manual)	Modifica el valor del objeto Manual_Usuario.
Set_Tiene_Ejemplar(\$Ejemplar)	Modifica el valor del atributo Tiene_Ejemplar.
Set_Obj_Disenio(\$aObj)	Modifica el Objeto Diseno.
Set_Obj_Marca(\$aObj)	Modifica el Objeto Marca.
Set_Obj_Tipo_Producto(\$ald)	Modifica el valor del atributo Tipo_Producto
Set_Obj_Proyecto(\$aObj)	Modifica el valor del atributo Proyecto

Nombre: EContrato	
Tipo de clase: Entidad	
Atributo	Tipo
Id_Contrato	Integer
Numero_Contrato	String
Descripción	String
Monto_Contrato	Integer
Vigencia_Contrato	String
Obj_Tipo_Contrato	Objeto Tipo de Contrato
Obj_Forma_Pago	Objeto Forma de Pago
Nombre	Descripción
EContrato(\$aNumeroContrato, \$aDescripcion, \$aMontoContrato, \$aVigenciaContrato, \$Obj_Tipo_Contrato, \$Obj_Forma_Pago)	Constructor de la clase.
Get_Id_Contrato()	Retorna el id del Contrato.
Get_Numero_Contrato()	Retorna el número de Contrato.
Get_Descripcion()	Retorna la descripción del Contrato.
Get_Monto_Contrato()	Retorna el monto del Contrato.
Get_Vigencia_Contrato()	Retorna la vigencia del Contrato.
Get_Obj_Tipo_Contrato()	Retorna el objeto Tipo de Contrato.
Get_Obj_Forma_Pago()	Retorna el objeto Forma de Pago.
Set_Id_Contrato(\$aldContrato)	Modifica el id del Contrato.
Set_Numero_Contrato(\$aNum)	Modifica el número de Contrato.
Set_Descripcion(\$aDescp)	Modifica la descripción del Contrato.
Set_Monto_Contrato(\$aMonto)	Modifica el monto del Contrato.
Set_Vigencia_Contrato(\$aVigencia)	Modifica la vigencia del Contrato.
Set_Obj_Tipo_Contrato(\$aObj)	Modifica el tipo de Contrato.
Set_Obj_Forma_Pago(\$aObj)	Modifica la forma de pago.

3.3.2.2 Descripción de Clases de Acceso a Datos.

Nombre: ADCliente	
Tipo de clase: Acceso a Datos	
Nombre	Descripción
ADCliente()	Constructor de la clase, en este caso invocando al constructor de la clase padre Base_Datos.
Insertar_Cliente(\$Obj_Cliente)	Inserta los datos del cliente en la tabla tb_cliente de la Base de Datos.
Modificar_Datos(\$Obj_Cliente)	Modifica los datos de un cliente en la tabla tb_cliente
No_Existen(\$Obj_Cliente)	Verifica si un cliente existe en la tabla



	tb_cliente.
Tiene_Datos_Economicos(Obj_Cliente)	Verifica si un cliente tiene los datos de las cuentas bancarias.
Cargar_Lista_Clientes(\$sql)	Retorna los clientes existentes en la base de datos de acuerdo a la consulta SQL que se le pase.
Cargar_Clientes_Combobox_Contratacion()	Retorna todos los clientes existentes en la base de datos.
Cargar_Datos_Cliente(\$ald_Cliente)	Retorna los datos del cliente con el id que se le pasa como parámetro.
Eliminar_Clientes(\$Arreglo_Id_Clientes)	Elimina los clientes cuyos id se encuentren en el arreglo que se le pasa como parámetro.

Nombre: ADCliente_Nacional	
Tipo de clase: Acceso a Datos	
Nombre	Descripción
ADCliente_Nacional()	Constructor de la clase, en este caso invocando al constructor de la clase padre ADCliente.
Insertar_Cliente_Nacional(\$Obj_Cliente)	Inserta los datos del cliente en la tabla tb_cliente y en la tabla tb_cliente_nacional.
Modificar_Cliente_Nacional(\$Obj_Cliente_Nacional_Actual, \$Obj_Cliente_Anterior)	Modifica los datos de un cliente.
Insertar_Cliente_Nacional_Especifico(\$Obj_Cliente)	Inserta en la tabla tb_cliente_nacional solamente.
Eliminar_Cliente_Nacional_Especifico(\$Obj_Cliente)	Elimina un cliente de la tabla tb_cliente_nacional solamente.

Nombre: ADCliente_Extranjero	
Tipo de clase: Acceso a Datos	
Nombre	Descripción
ADCliente_Extranjero()	Constructor de la clase, en este caso invocando al constructor de la clase padre ADCliente.
Insertar_Cliente_Extranjero(\$Obj_Cliente)	Inserta los datos del cliente en la tabla tb_cliente y en la tabla tb_cliente_extranjero.
Modificar_Cliente_Extranjero(\$Obj_Cliente_Extranjero_Actual, \$Obj_Cliente_Anterior, \$ADObj_Representante_Extranjero, \$ADObj_Representante_Nacional, \$ADObj_Cuenta_Bancaria, \$ADObj_Cliente_Nacional)	Modifica los datos de un cliente.
Insertar_Cliente_Extranjero_Especifico(\$Obj_Cliente)	Inserta en la tabla tb_cliente_extranjero solamente.
Eliminar_Cliente_Extranjero_Especifico(\$Obj_Cliente)	Elimina un cliente de la tabla



iente)	tb_cliente_extranjero solamente.
--------	----------------------------------

Nombre: ADProyecto	
Tipo de clase: Acceso a Datos	
Nombre	Descripción
ADProyecto()	Constructor de la clase
No_Existe(\$Obj_Proyecto)	Verifica si existe un proyecto en la base de datos
Modificar_Datos_Proyecto(\$Obj_Proyecto)	Modifica los datos del proyecto
Insertar_Proyecto(EProyecto \$Obj_Proyecto)	Inserta el proyecto
Cargar_Lista_Proyectos(\$sql)	Carga la lista de proyectos
Eliminar_Proyectos(\$Arreglo_Id_Proyectos)	Elimina un proyecto
Obtener_Lista_Proyectos(\$sql)	Retorna una lista de proyectos
Cargar_Datos(\$ald_Proyecto)	Retorna los datos de un proyecto
Poner_Valor_Id_Contrato_Proyecto(\$lista_proyectos_vincular, \$Id_Contrato)	Vincula proyectos a contratos
Eliminar_Cronograma_Ejecucion(\$EObj_Proyecto)	Elimina el cronograma de ejecución
Eliminar_Proyecto_Tecnico(\$EObj_Proyecto)	Elimina el proyecto técnico.

Nombre: ADProducto	
Tipo de clase: Acceso a Datos	
Nombre	Descripción
ADProducto()	Constructor de la clase
No_Existe(\$Obj_Producto)	Verifica si existe un producto en la base de datos
Cargar_Datos(\$ald_Producto)	Retorna los datos de los productos
Insertar_Producto(EProducto \$Obj_Producto)	Inserta un producto en la BD
Modificar_Producto(EProducto \$Obj_Producto, EProducto \$Obj_Producto_Anterior)	Modifica los datos de un producto
Obtener_Herramientas_Utilizadas(\$ald_Producto)	Retorna las herramientas que se utilizaron para desarrollar el producto.
Obtener_Lenguajes_Utilizados(\$ald_Producto)	Retorna los lenguajes utilizados
Obtener_Lista_Productos(\$sql)	Retorna una lista de productos
Eliminar_Productos(\$Arreglo_Id_Productos)	Elimina un producto

Nombre: ADContrato	
Tipo de clase: Acceso a Datos	
Nombre	Descripción
ADContrato()	Constructor de clase.
No_Existe_Numero_Contrato(EContrato)	Verifica si existe un contrato.

\$EObj_Contrato)	
Insertar_Contrato(EContrato \$EObj_Contrato)	Inserta un contrato
Modificar_Contrato(EContrato \$Obj_Contrato)	Modifica un contrato
Cargar_Lista_Contratos(\$sql)	Retorna una lista de contratos
Cargar_Datos_Contrato(\$ald_Contrato)	Retorna los datos de un contrato
Cargar_Datos_Interes(\$al_Contrato)	Retorna datos específicos
Eliminar_Contratos(\$Arreglo_Id_Contrato)	Elimina contratos
Vincular_Proyectos_Contrato(\$lista_proyectos_vincular, \$Id_Contrato)	Vincula proyectos a contratos
Subir_Suplementos(ESuplementosContrato \$EObj_Suplemento)	Coloca archivos en el sitio

3.3.2.3 Descripción de la Clase Fachada.

Nombre: Fachada	
Tipo de clase: Controladora	
Atributo	Tipo
\$Obj_MGestionar_Producto	Objeto
\$Obj_MGestionar_Cliente	Objeto
\$Obj_MGestionar_Proyecto	Objeto
\$Obj_MGestionar_Contrato	Objeto
Nombre	Descripción
Fachada()	Constructor de la clase.
Get_Obj_MGestionar_Producto()	Devuelve el objeto producto.
Set_Obj_MGestionar_Producto(\$Obj)	Modifica el objeto Producto.
Get_Obj_MGestionar_Cliente()	Devuelve el objeto cliente.
Set_Obj_MGestionar_Cliente(\$Obj)	Modifica el objeto cliente.
Get_Obj_MGestionar_Proyecto()	Devuelve el objeto proyecto.
Set_Obj_MGestionar_Proyecto(\$Obj)	Modifica el objeto proyecto.
Get_Obj_MGestionar_Contrato()	Devuelve el objeto contrato.
Set_Obj_MGestionar_Contrato(\$Obj)	Modifica el objeto contrato.
Obtener_Listado_Productos(\$aNomb_Prod, \$aNomb_Proyecto, \$aTipo_Proyecto, \$aTipo_Producto, \$aEmpresa_Pertenece, \$aSiglas_Pertenece, \$aNumContrato, \$ald_Producto=null, \$ald_Proyecto=null)	Retorna el listado de productos de acuerdo a los parámetros que se le pasen.
Adicionar_Cliente_Nacional(\$aNombre_Cliente, \$aSiglas_Cliente, \$aResolucion_Creacion, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante, \$aApellidos_Representante, \$aResolucion_Nombramiento, \$aCuenta_Cuc,	Inserta un cliente de tipo nacional en la base de datos.



\$ald_Banco_Cuc, \$aSucursal_Cuc, \$aCuenta_Cup, \$ald_Banco_Cup)	
Adicionar_Cliente_Extranjero(\$aTipo_Cliente, \$ald_Pais_Cliente,\$aNOMBRE_Cliente, \$aSiglas_Cliente, \$aActa_Representacion, \$aNumero_Registro, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNOMBRE_Representante, \$aApellidos_Representante, \$aActa_Contitucion, \$aCuenta, \$ald_Tipo_Moneda, \$ald_Banco, \$aSucursal)	Inserta un cliente extranjero en la base de datos.
Obtener_Lista_Clientes(\$aNomb_Cliente=null, \$aSig_Cliente=null, \$aTipo_Cliente=null)	Retorna una lista de clientes de acuerdo a los parámetros que se le pasen
Obtener_Lista_Proyectos(\$aNomb_Proyecto, \$aTipo_Proyecto, \$aNomb_Cliente_pertenece, \$aSig_Cliente_Pertenece, \$Num_Contrato_Pertenece, \$ald_Proyecto=null)	Retorna una lista de proyectos de acuerdo a los parámetros que se le pasen.
Obtener_Lista_Contratos(\$aNumero_Contrato, \$aNomb_Cliente_Pertenece, \$aSig_Cliente_Pertenece)	Retorna una lista de contratos de acuerdo a los parámetros que se le pasen.
Obtener_Datos_Cliente(\$ald)	Retorna los datos del cliente cuyo id se le pasa como parámetro.
Eliminar_Clientes(\$aLista_Id_Clientes)	Elimina los clientes que coincidan con los id que se le pasan en la lista.
Eliminar_Proyectos(\$aLista_Id_Proyectos)	Elimina los proyectos cuyos id coincidan con los que se pasan en la lista.
Modificar_Datos_Cliente_Nacional(ECliente \$Obj_Cliente_Anterior, \$aNOMBRE_Cliente, \$aSiglas_Cliente, \$aResolucion_Creacion, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNOMBRE_Representante, \$aApellidos_Representante, \$aResolucion_Nombramiento, \$aCuenta_Cuc, \$ald_Banco_Cuc, \$Nombre_Banco_Cuc, \$aSucursal_Cuc, \$aCuenta_Cup, \$ald_Banco_Cup, \$Nombre_Banco_Cup)	Modifica los datos de un cliente nacional.
Modificar_Datos_Cliente_Extranjero(ECliente \$Obj_Cliente_Anterior, \$aTipo_Cliente, \$ald_Pais_Cliente, \$Nombre_Pais, \$aNOMBRE_Cliente, \$aSiglas_Cliente, \$aActa_Representacion, \$aNumero_Registro, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNOMBRE_Representante, \$aApellidos_Representante, \$aActa_Contitucion, \$aCuenta, \$ald_Tipo_Moneda, \$Nombre_Tipo_Moneda, \$ald_Banco, \$Nombre_Banco, \$aSucursal)	Modifica los datos de un cliente extranjero.
Insertar_Proyecto(\$aTipo_Proyecto, \$aNOMBRE_Proyecto, ECliente \$Obj_Cliente, \$aNOMBRE_Responsable_Cliente,	Inserta un proyecto en la base de datos.



\$aApellidos_Responsable_Cliente, \$aCorreo_Responsable_Cliente, \$aNombre_Responsable_Uci, \$aApellidos_Responsable_Uci, \$aCorreo_Responsable_Uci, \$ald_Area_Uci, \$Proyecto_Tecnico, \$aTamano_Proyecto_Tecnico, \$alIdentificador_Proyecto_Tecnico, \$Cronograma_Ejecucion, \$aTamano_Cronograma_Ejecucion, \$alIdentificador_Cronograma_Ejecucion)	
Modificar_Proyecto(EProyecto \$EObj_Proyecto_Anterior, \$aTipo_Proyecto, \$aNombre_Proyecto, \$aNombre_Responsable_Cliente, \$aApellidos_Responsable_Cliente, \$aCorreo_Responsable_Cliente, \$aNombre_Responsable_Uci, \$aApellidos_Responsable_Uci, \$aCorreo_Responsable_Uci, \$ald_Area_Uci, \$aNombre_Area, \$Proyecto_Tecnico, \$aTamano_Proyecto_Tecnico, \$alIdentificador_Proyecto_Tecnico, \$Cronograma_Ejecucion, \$aTamano_Cronograma_Ejecucion, \$alIdentificador_Cronograma_Ejecucion)	Modifica los datos de un proyecto.
Obtener_Datos_Proyecto(\$ald_Proyecto)	Obtiene los datos de un proyecto pasándole como parámetro el id.
Insertar_Contrato(\$NumeroContrato, \$Descripcion, \$MontoContrato, \$VigenciaContrato, \$Id_Tipo_Contrato, \$Tipo_Forma_Pago, \$aMonto_Anticipo, \$Id_Tipo_Transferencia, \$Matriz_Datos_Porcientos, \$lista_proyectos_vincular)	Inserta un contrato en la base de datos.
Obtener_Datos_Contrato(\$ald_Contrato)	Retorna los datos de un contrato pasándole el id.
Modificar_Contrato(EContrato \$EObj_Contrato_Anterior, \$NumeroContrato, \$Descripcion, \$MontoContrato, \$VigenciaContrato, \$Id_Tipo_Contrato, \$Nombre_Tipo_Contrato, \$Tipo_Forma_Pago, \$aMonto_Anticipo, \$Id_Tipo_Transferencia, \$Nombre_Tipo_Transferencia, \$Matriz_Datos_Porcientos)	Modifica los datos de un contrato.
Eliminar_Contratos(\$Arreglo_Id_Contrato)	Elimina los contratos cuyos id coincidan con los que se le pasen en el arreglo.
Insertar_Producto(\$Obj_Proyecto, \$aNombre_Producto, \$ald_Tipo_Producto, \$Nombre_Tipo_Producto, \$aCertificado_Calisoft, \$aEjemplar, \$Manual_Usuario, \$aTamano_Manual_Usuario,	Inserta un producto en la base de datos.

\$alIdentificador_Manual_Usuario)	
Obtener_Datos_Producto(\$ald_Producto)	Retorna los datos de un producto pasándole el id.
Modificar_Datos_Producto(EProducto \$Obj_Producto_Anterior, \$aNOMBRE_Producto, \$ald_Tipo_Producto, \$Nombre_Tipo_Producto, \$aCertificado_Calisoft, \$aEjemplar, \$Manual_Usuario, \$aTamano_Manual_Usuario, \$alIdentificador_Manual_Usuario)	Modifica los datos de un producto.
Eliminar_Productos(\$Arreglo_Id_Productos)	Elimina los productos cuyos id es ten en el arreglo que se pasa como parámetro.

3.3.2.4 Descripción de la Clase Fachada de Acceso a Datos.

Nombre: Fachada de Acceso a Datos	
Tipo de clase: Controladora	
Atributo	Tipo
\$Obj_ADProducto	Objeto de tipo ADProducto.
\$Obj_ADContrato	Objeto de tipo ADContrato.
\$Obj_ADCliente	Objeto de tipo ADCliente.
\$Obj_ADCliente_Nacional	Objeto de tipo ADCliente Nacional.
\$Obj_ADCliente_Extranjero	Objeto de tipo ADCliente Extranjero.
\$Obj_ADProyecto	Objeto de tipo ADProyecto.
Nombre	Descripción
Get_Obj_ADProducto()	Retorna el Objeto de tipo ADProducto.
Get_Obj_ADContrato()	Retorna el Objeto de tipo ADContrato.
Get_Obj_ADCliente()	Retorna el Objeto de tipo ADCliente.
Get_Obj_ADCliente_Nacional()	Retorna el Objeto de tipo ADCliente Nacional.
Get_Obj_ADCliente_Extranjero()	Retorna el Objeto de tipo ADCliente Extranjero.
Get_Obj_ADProyecto()	Retorna el Objeto de tipo ADProyecto.
Obtener_Datos_Producto(\$ald_Producto)	Retorna los datos de un producto pasándole el id.
Insertar_Producto(EProducto \$EObj_Producto)	Inserta un producto en la base de datos.
Modificar_Datos_Producto(EProducto \$EObj_Producto)	Modifica los datos de un proyecto.
Eliminar_Productos(\$Arreglo_Id_Productos)	Elimina una lista de productos.
Cargar_Lista_Productos(\$aNomb_Prod, \$aNomb_Proyecto, \$aTipo_Proyecto, \$aTipo_Producto, \$aCliente_Pertenece, \$aSiglas_Cliente_Pertenece, \$aNumContrato, \$ald_Producto=null, \$ald_Proyecto=null)	Carga una lista de productos.
Obtener_Datos_Contrato(\$ald_Contrato)	Retorna los datos de un contrato.



Eliminar_Contratos(\$Arreglo_Id_Contrato)	Elimina una lista de contratos.
Modificar_Datos_Contrato(EContrato \$EObj_Contrato, EContrato \$EObj_Anterior)	Modifica los datos de un contrato.
Insertar_Contrato(EContrato \$EObj_Contrato, \$Lista_proyectos_vincular=null)	Inserta un contrato en la base de datos.
Cargar_Lista_Contratos(\$aNumero_Contrato, \$aNomb_Cliente_Pertenece, \$aSig_Cliente_Pertenece)	Retorna una lista de contratos.
Eliminar_Clientes(\$aLista_Id_Clientes)	Elimina una lista de clientes.
Cargar_Lista_Clientes(\$aNomb_Cliente, \$aSig_Cliente, \$aTipo_Cliente)	Retorna una lista de clientes.
Obtener_Datos_Cliente(\$ald_Cliente)	Retorna los datos de un cliente.
Insertar_Cliente_Nacional(ECliente_Nacional \$Obj_Cliente)	Inserta un cliente en la base de datos.
Modificar_Cliente_Nacional(ECliente \$EObj_Cliente_Anterior, ECliente_Nacional \$Obj_Cliente)	Modifica los datos de un cliente nacional.
Modificar_Cliente_Extranjero(ECliente \$EObj_Cliente_Anterior, ECliente_Extranjero \$Obj_Cliente)	Modifica los datos de un cliente extranjero.
Insertar_Cliente_Extranjero(ECliente_Extranjero \$Obj_Cliente)	Inserta un cliente extranjero.
Insertar_Proyecto(EProyecto \$EObj_Proyecto)	Inserta un proyecto en la base de datos.
Modificar_Datos_Proyecto(EProyecto \$EObj_Proyecto)	Modifica los datos de un proyecto.
Eliminar_Proyectos(\$aLista_Id_Proyectos)	Elimina una lista de proyectos.
Cargar_Lista_Proyectos(\$aNomb_Proyecto, \$aTipo_Proyecto, \$aNomb_Cliente_pertenece, \$aSig_Cliente_Pertenece, \$Num_Contrato_Pertenece, \$Id_Proyecto=null)	Retorna una lista de proyectos.
Cargar_Datos_Proyecto(\$ald_Proyecto)	Retorna los datos de un proyecto.

3.3.2.5 Descripción de Clases Manager.

Nombre: Manager Cliente	
Tipo de clase: Controladora	
Atributo	Tipo
\$Obj_Fachada_Acceso_Dato	Objeto de tipo Fachada de Acceso a Datos.
Nombre	Descripción
Get_Obj_Fachada_Acceso_Dato()	Retorna el Objeto de tipo Fachada de Acceso a Datos.
Insertar_Cliente_Nacional(\$aNombre_Cliente, \$aSiglas_Cliente, \$aResolucion_Creacion, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante,	Inserta un cliente en la base de datos.



<p>\$aApellidos_Representante, \$aResolucion_Nombramiento, \$aCuenta_Cuc, \$ald_Banco_Cuc, \$aSucursal_Cuc, \$aCuenta_Cup, \$ald_Banco_Cup)</p>	
<p>Devolver_Cliente_Nacional(\$aTipo_Cliente, \$aNombre_Cliente, \$aSiglas_Cliente, \$aResolucion_Creacion, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante, \$aApellidos_Representante, \$aResolucion_Nombramiento, \$aCuenta_Cuc, \$ald_Banco_Cuc, \$aSucursal_Cuc, \$aCuenta_Cup, \$ald_Banco_Cup)</p>	<p>Retorna los datos de un cliente nacional de acuerdo a los parámetros que se le pasen.</p>
<p>Insertar_Cliente_Extranjero(\$aTipo_Cliente, \$ald_Pais_Cliente,\$aNombre_Cliente, \$aSiglas_Cliente, \$aActa_Representacion, \$aNumero_Registro, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante, \$aApellidos_Representante, \$aActa_Contitucion, \$aCuenta, \$ald_Tipo_Moneda, \$ald_Banco, \$aSucursal)</p>	<p>Inserta un cliente extranjero en la base de datos.</p>
<p>Devolver_Cliente_Extranjero(\$aTipo_Cliente, \$ald_Pais_Cliente,\$aNombre_Cliente, \$aSiglas_Cliente, \$aActa_Representacion, \$aNumero_Registro, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante, \$aApellidos_Representante, \$aActa_Contitucion, \$aCuenta, \$ald_Tipo_Moneda, \$ald_Banco, \$aSucursal)</p>	<p>Devuelve los datos de un cliente extranjero.</p>
<p>Modificar_Cliente_Extranjero(ECliente \$Obj_Cliente_Anterior, \$aTipo_Cliente, \$ald_Pais_Cliente, \$Nombre_Pais, \$aNombre_Cliente, \$aSiglas_Cliente, \$aActa_Representacion, \$aNumero_Registro, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante, \$aApellidos_Representante, \$aActa_Contitucion, \$aCuenta, \$ald_Tipo_Moneda, \$Nombre_Tipo_Moneda, \$ald_Banco, \$Nombre_Banco, \$aSucursal)</p>	<p>Modifica los datos de un cliente extranjero.</p>
<p>Modificar_Cliente_Nacional(ECliente \$Obj_Cliente_Anterior, \$aNombre_Cliente, \$aSiglas_Cliente, \$aResolucion_Creacion, \$aObjeto_Social_Cliente, \$aFax_Cliente, \$aTelefono_Cliente, \$aNombre_Representante, \$aApellidos_Representante, \$aResolucion_Nombramiento, \$aCuenta_Cuc, \$ald_Banco_Cuc, \$Nombre_Banco_Cuc,</p>	<p>Modifica los datos de un cliente nacional.</p>



\$aSucursal_Cuc, \$aCuenta_Cup, \$ald_Banco_Cup, \$Nombre_Banco_Cup)	
Cargar_Datos_Cliente(\$ald)	Retorna los datos de un cliente, pasándole el id.
Eliminar_Clientes(\$aLista_Id_Clientes)	Elimina una lista de clientes.

Nombre: Manager Proyecto	
Tipo de clase: Controladora	
Atributo	Tipo
\$Obj_Fachada_Acceso_Dato	Objeto de tipo Fachada de Acceso a Datos.
Nombre	Descripción
Get_Obj_Fachada_Acceso_Dato()	Retorna el Objeto de tipo Fachada de Acceso a Datos.
Insertar_Proyecto(\$aTipo_Proyecto, \$aNombre_Proyecto, \$Obj_Proyecto_Tecnico, \$Obj_Cronograma_Ejecucion, ECliente \$Obj_Cliente, \$aNombre_Responsable_Cliente, \$aApellidos_Responsable_Cliente, \$aCorreo_Responsable_Cliente, \$aNombre_Responsable_Uci, \$aApellidos_Responsable_Uci, \$aCorreo_Responsable_Uci, \$ald_Area_Uci)	Inserta un proyecto en la base de datos.
Cargar_Lista_Proyectos(\$aNomb_Proyecto, \$aTipo_Proyecto, \$aNomb_Cliente_pertenece, \$aSig_Cliente_Pertenece, \$Num_Contrato_Pertenece, \$Id_Proyecto=null)	Retorna una lista de proyectos en la base de datos.
Cargar_Datos_Proyecto(\$ald_Proyecto)	Retorna los datos de un proyecto.
Modificar_Datos_Proyecto(EProyecto \$EObj_Proyecto_Anterior, \$aTipo_Proyecto, \$aNombre_Proyecto, \$Obj_Proyecto_Tecnico, \$Obj_Cronograma_Ejecucion, \$aNombre_Responsable_Cliente, \$aApellidos_Responsable_Cliente, \$aCorreo_Responsable_Cliente, \$aNombre_Responsable_Uci, \$aApellidos_Responsable_Uci, \$aCorreo_Responsable_Uci, \$ald_Area_Uci, \$aNombre_Area)	Modifica los datos de un proyecto.
Eliminar_Proyectos(\$aLista_Id_Proyectos)	Elimina una lista de proyectos

Nombre: Manager Contrato.	
Tipo de clase: Controladora	
Atributo	Tipo
\$Obj_Fachada_Acceso_Dato	Objeto de tipo Fachada de Acceso a Datos.



Nombre	Descripción
Get_Obj_Fachada_Acceso_Dato()	Retorna el Objeto de tipo Fachada de Acceso a Datos.
Devolver_Obj_Contrato(\$NumeroContrato, \$Descripcion, \$MontoContrato, \$VigenciaContrato, \$Id_Tipo_Contrato, \$Tipo_Forma_Pago, \$aMonto_Anticipo, \$Id_Tipo_Transferencia, \$Matriz_Datos_Porcientos)	Retorna un objeto de tipo contrato de acuerdo a los parámetros que se pasan.
Cargar_Lista_Contratos(\$aNumero_Contrato, \$aNomb_Cliente_Pertenece, \$aSig_Cliente_Pertenece)	Retorna una lista de contratos.
Cargar_Datos_Contrato(\$ald_Contrato)	Retorna los datos de un contrato.
Modificar_Datos_Contrato(EContrato \$EObj_Contrato_Anterior, \$NumeroContrato, \$Descripcion, \$MontoContrato, \$VigenciaContrato, \$Id_Tipo_Contrato, \$Nombre_Tipo_Contrato, \$Tipo_Forma_Pago, \$aMonto_Anticipo, \$Id_Tipo_Transferencia, \$Nombre_Tipo_Transferencia, \$Matriz_Datos_Porcientos)	Modifica los datos de un contrato.
Eliminar_Contratos(\$Arreglo_Id_Contrato)	Eliminar una lista de contratos.

Nombre: Manager Producto.	
Tipo de clase: Controladora	
Atributo	Tipo
\$Obj_Fachada_Acceso_Dato	Objeto de tipo Fachada de Acceso a Datos.
Nombre	Descripción
Get_Obj_Fachada_Acceso_Dato()	Retorna el Objeto de tipo Fachada de Acceso a Datos.
Obtener_Lista_Productos(\$aNomb_Prod, \$aNomb_Proyecto, \$aTipo_Proyecto, \$aTipo_Producto, \$aCliente_Pertenece, \$aSiglas_Cliente_Pertenece, \$aNumContrato, \$ald_Producto=null, \$ald_Proyecto=null)	Retorna una lista de productos.
Insertar_Producto(\$Obj_Proyecto, \$aNombre_Producto, \$ald_Tipo_Producto, \$Nombre_Tipo_Producto, \$aCertificado_Calisoft, \$aEjemplar, \$Obj_Archivo_Manual_Usuario)	Inserta un producto en la base de datos.
Obtener_Datos_Producto(\$ald_Producto)	Retorna los datos de un producto.
Modificar_Datos_Producto(EProducto \$Obj_Producto_Anterior, \$Nombre_Producto, \$Id_Tipo_Producto, \$Nombre_Tipo_Producto, \$Certificado_Calisoft, \$Ejemplar, \$Obj_Archivo_Manual_Usuario)	Modifica los datos de un producto.
Eliminar_Productos(\$Arreglo_Id_Productos)	Elimina una lista de productos.

3.3.3 Diagrama de Clases Persistentes.

Las clases persistentes son aquellas que la información que encapsulan persisten en el tiempo, o sea, son comúnmente las clases que se mapean contra las tablas de la base de datos. En el Anexo 3 se muestran los diagramas de clases persistentes ya que debido a su extensión se ha tenido que dividir en tres partes.

3.3.4 Diseño de la Base de Datos.

En el diagrama de base de datos se representa la distribución física de las tablas en la base de datos, mostrando además los atributos de las mismas. En el Anexo 4 se encuentra el diagrama de la Base de Datos debido a su extensión, el cual se encuentra dividido en seis partes por la cantidad de tablas que presenta.

3.3.5 Patrones de Diseño.

3.3.5.1 Patrón Fachada.

Una de las principales etapas dentro del diseño del software, es el diseño de la arquitectura y dentro de la misma, la actividad más importante es la determinación de subsistemas que agrupen, con un alto nivel de cohesión, a un grupo determinado de clases del diseño relacionadas entre sí o que participen en determinados casos de uso. Un Subsistema al igual que la clase posee una interfaz y encapsula algo. Una clase encapsula estado y comportamiento y oculta esta información tras una interfaz pública de acceso a sus servicios. Asimismo un subsistema ideal debería encapsular las clases participantes y las colaboraciones entre las mismas y ocultarlas tras una interfaz de acceso a los servicios proporcionados por el subsistema (FERNÁNDEZ, 2007).

Precisamente el **Patrón Fachada** trata de ayudar a conseguir todos estos objetivos de diseño arquitectónico, dotando a los subsistemas de una interfaz de acceso análoga a la que proporciona una clase. Su objetivo principal es proporcionar una interfaz unificada de alto nivel para un subsistema, que oculta las interfaces de bajo nivel de las clases que lo implementan. Con esto se consiguen dos objetivos fundamentales: hacer el subsistema más fácil de usar y desacoplar a los clientes de las clases del subsistema (FERNÁNDEZ, 2007).

En correspondencia con lo explicado anteriormente, la estructura del patrón Fachada quedaría como se muestra a continuación:

- **Clases del Subsistema:** Implementan la funcionalidad del subsistema, tienen un estrecho conocimiento las unas de las otras y colaboran para llevar a cabo el trabajo asignado por la Fachada y además no tienen ningún conocimiento de la existencia del objeto Fachada.
- **Fachada:** Conoce la distribución de responsabilidades entre las clases del subsistema. Mantiene referencias hacia muchos de los objetos del subsistema y delega en ellos para llevar a cabo su trabajo. No tiene ningún conocimiento acerca de los clientes.
- **Clientes:** Mantienen una referencia al objeto Fachada. Envían sus mensajes al mismo para acceder a la funcionalidad del subsistema. En general no mantienen referencias a los objetos internos del subsistema, pero no se les impide adquirirlas si lo necesitan (FERNÁNDEZ, 2007).

Teniendo en cuenta la estructuración del sistema en cuestión se ha decidido adoptar la siguiente estructura basándonos en el patrón Fachada:

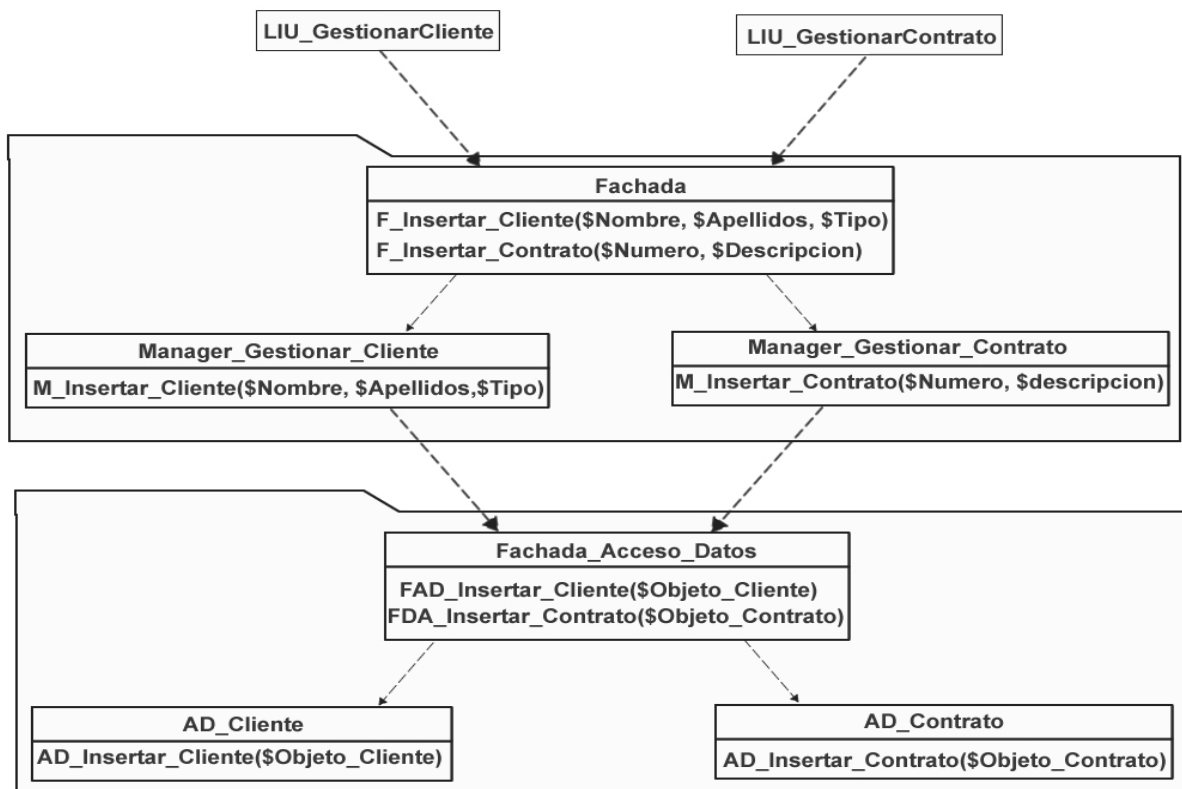


Figura 3.1 Uso del Patrón Fachada.

Para cualquiera de los casos la utilización de una Fachada incluye la característica unidireccional, lo que se traduce en que la fachada instancia y conoce objetos de las clases que encapsula pero las clases encapsuladas en ningún momento tienen conocimiento de la existencia de la misma.

3.3.5.2 Patrón Objeto de Acceso a Datos (DAO).

El patrón DAO (Data Access Object) es una solución al problema del diferencial de impedancia entre un programa de aplicación orientado a objetos y una base de datos relacional, empleando únicamente la interfaz de programación nativa del manejador de base de datos (CEYUSA 2007).

El patrón DAO forma parte del catálogo de patrones para J2EE, pero como cualquier patrón, es perfectamente adaptable a cualquier lenguaje de programación (CEYUSA 2007).

El Objeto de Acceso a Datos, se encargará entonces de conectarse directamente con la fuente de datos utilizando su interfaz de programación (API) nativa y ejecutará las transacciones necesarias sobre ella. Por ejemplo, si la operación es una consulta, el objeto hará la transacción indicada y devolverá a la aplicación un conjunto de objetos de transporte de datos conteniendo el resultado; por otro lado, si la transacción es una inserción, entonces la aplicación pasará como parámetro, un objeto de transporte de datos que será almacenada en la fuente de datos, y así con las demás tareas como borrado y actualización (CEYUSA 2007).

Para el caso que nos ocupa, se implementará una clase DAO por cada clase Entidad implementada para la solución del problema. De esta forma garantizará el buen funcionamiento del sistema y la presencia de tres ventajas fundamentales.

- Se tiene un atenuante al problema del diferencial de impedancia (transparencia).
- Se baja en nivel de acoplamiento entre clases, reduciendo la complejidad de realizar cambios
- Se aísla las conexiones a la fuente de datos en una capa fácilmente identificable y mantenible.

3.3.6 Definiciones de diseño.

El diseño es un aspecto muy importante para cualquier ámbito en el que se presente, el mismo debe primero que todo, basar sus características en el usuario con el cual interactúa. En este

caso en particular son estudiantes, profesores y trabajadores de la Universidad, los cuales en ocasiones no tienen amplios conocimientos de Informática, por ello, nos hemos apoyado en varios principios de diseño universal que garantizan el correcto aprovechamiento del mismo.

Uso Equiparable: se deben facilitar las mismas maneras de uso para cada usuario en particular, se debe evitar segregar o estigmatizar a cualquier usuario, deben estar igualmente disponibles Las características de privacidad, garantía y seguridad para cada usuario.

Uso Flexible: ofrecer posibilidades de elección en los métodos de uso, exactitud y precisión, y se adapte al paso o ritmo del usuario.

Simple e Intuitivo: debe eliminar complejidades innecesarias, ser consistente con las expectativas e intuición del usuario, que presente la información acorde con su importancia y que proporcione avisos eficaces durante la realización de una tarea.

Información perceptible: proporcionar contraste suficiente entre la información esencial y sus alrededores, ampliar la legibilidad de la información esencial.

Con tolerancia al error: proporcionar advertencias sobre peligros y errores, proporcionar características seguras de interrupción, disponer de elementos para minimizar los riesgos y errores.

Que exija poco esfuerzo físico: permitir que el usuario mantenga una posición corporal neutra, minimizar las acciones repetitivas, minimizar el esfuerzo físico continuado.

Tamaño y espacio para el acceso y uso: proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie, el alcance de cualquier componente debe ser confortable para cualquier usuario sentado o de pie.

3.3.7 Tratamiento de errores.

Los errores son tratados de dos formas diferentes y la relación entre ambos tiene como objetivo el correcto funcionamiento del Sistema. La primera tiene lugar en la Página Cliente con la utilización del lenguaje JavaScript, el cual brinda la posibilidad de mostrar mensajes de alerta embebidos en la página (Ver Figura 3.2), los cuales son más aceptados actualmente por los usuarios. Con esto se evita, la entrada de datos erróneos a las páginas servidoras y su ejecución nuevamente con datos inconsistentes. La segunda forma tiene lugar en las Páginas Servidoras, en caso de que se filtre algún dato erróneo, allí se analizará nuevamente los datos

manejados y se mostrarán mensajes embebidos en la Página Cliente (Ver Figura 3.2) informando las causas del mismo al Usuario que se encuentre logueado.

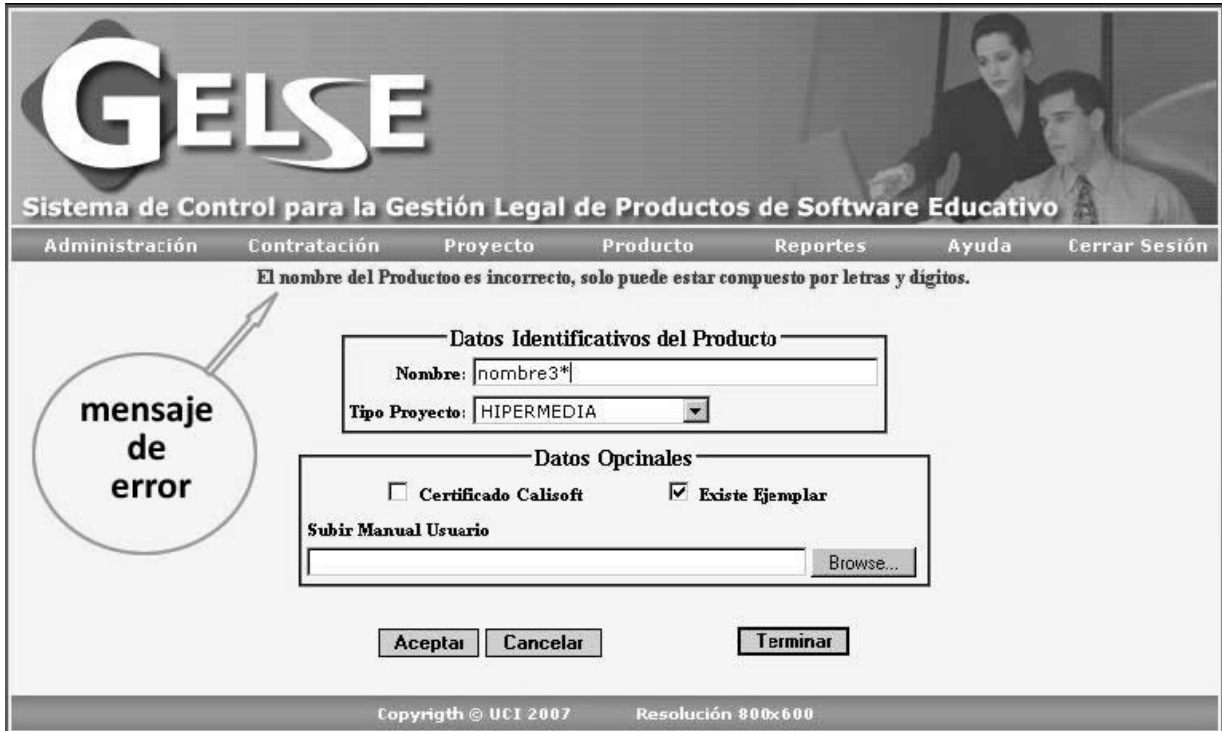


Figura 3.2 Mensaje de error.

3.3.8 Seguridad.

El tema de la seguridad se basa en el empleo de la autenticación como una acción de obligatorio cumplimiento para el uso de la aplicación. Para ello cada usuario tendrá un nombre y una contraseña la cual podrá ser modificada por dicho usuario o por el administrador de manera eventual con el objetivo de evitar un uso incorrecto del sistema. Además, solo el administrador podrá crear un nuevo usuario, y durante su creación asignarle el rol que desempeñará. Precisamente la utilización de roles es otra técnica que se utiliza, donde cada rol tendrá acceso a determinadas funcionalidades de acuerdo a su desempeño. El sistema además, implementa un mecanismo de encriptación utilizando MD5, lo cual brindará una mayor protección a la información que se maneja.

3.3.9 Interfaz.

La interfaz está confeccionada siguiendo el esquema Cabecera-Navegador-Contenido, con el objetivo de lograr un diseño consistente. En la cabecera se presenta en la esquina superior derecha el logotipo del Sistema, puesto que es ahí a donde primero dirige la vista el usuario

según estudios realizados. El navegador se presenta en forma de Menú Desplegable con el propósito de que el área de Contenido sea lo más amplia posible, puesto que el sistema maneja un gran número de información ya sea en forma de entrada o de reporte, y debe procesarse toda en una misma página. Ver Figura 3.3.

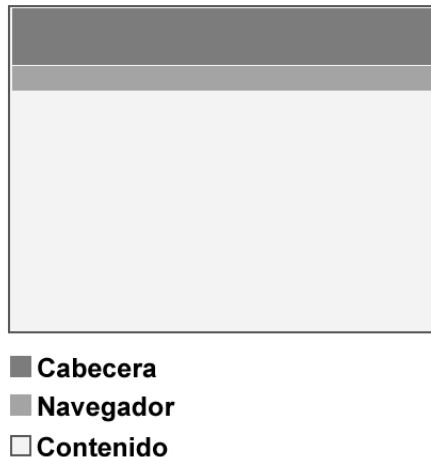


Figura 3.3 Esquema de la Interfaz

Se utilizan las tablas para mostrar la información que se maneja, así como para la ubicación de imágenes como el banner y demás componentes de la plantilla que se utilizan y porque además tanto las tablas como las plantillas son compatibles con cualquier navegador. Se utiliza por otro lado una hoja de estilos que ofrece colores, tamaño y tipo de fuente, en dependencia de la importancia de la información, tanto a tablas, vínculos, botones, como texto. Predominan tonalidades de color azul claro, combinadas con el color blanco para lograr una interfaz agradable a la vista del usuario.

3.3.10 Estándares de codificación.

El estándar de codificación que se utilizará para lograr la organización y un fácil entendimiento del código si en un futuro fuera necesario, es el siguiente:

3.3.10.1 Declaraciones:

Variables explícitas. Ejemplo: `var $Número_Contrato;`

La declaración de Clases se hará de la siguiente manera:

Para la clase Fachada General: Fachada.

Para las Clases Manager se pondrá una M mayúscula delante del nombre, el cual comenzará también con mayúscula. Ejemplo: M Gestionar Proyecto.

Para las Clases de Acceso a Datos se pondrá las Siglas AD y seguido el nombre comenzando con mayúscula. Ejemplo: AD Proyecto.

Para las Clases entidad se pondrá una E mayúscula delante del nombre, el cual también comenzará con mayúscula. Ejemplo: E Proyecto.

3.3.10.2 Indentación.

Es una tabulación donde existirán tres espacios en blanco después de la llave.

```
if($datos == -1)
{
    $this->Set_Error("No existe ese cliente.");
    Return false;
}
```

3.3.10.3 Espacios en blanco

Espacios en blanco entre operadores lógicos-aritméticos y sus operandos.

```
if($datos == -1)
```

3.3.11 Concepción de la ayuda.

Debido a que la aplicación va dirigida a un personal que en la mayoría de los casos tendrán al menos un nivel mínimo de conocimientos de informática, se ha decidido desarrollar una ayuda que se centre en explicar de manera formal las acciones que se podrán realizar al seleccionar una opción en específico. Ejemplo: dentro de las acciones relacionadas con el Contrato están incluidas las de los Anexos al Contrato.

3.4 Conclusiones del Capítulo.

En este Capítulo se desarrolló la modelación del sistema empleando para ello, los diagramas de clases del análisis y de diseño, dejando bien definida la lógica del negocio en cuestión. Se realiza una breve descripción de la responsabilidad de las clases controladoras inmersas en el sistema. Se confecciona el diagrama de clases persistentes y el modelo de datos. Se exponen las cuestiones referentes a los patrones de diseño, el tratamiento de errores, las características de la interfaz, la seguridad que ofrecerá el sistema y se describe la estructura de la ayuda.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente Capítulo se muestra el diagrama de despliegue en el cual se representan los nodos del sistema, se basaron para esto en las características y funcionalidades que debe cumplir el sistema. También se muestra en el modelo de implementación los diagramas de los componentes que contiene la aplicación. Por último, se presentan las pruebas realizadas al sistema con el objetivo de corroborar el correcto funcionamiento del mismo.

4.2 Modelo de Despliegue.

En el modelo de despliegue se representa la configuración de los tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes. En el mismo se muestra una PC Cliente desde la cual el usuario podrá utilizar la aplicación, una PC Servidor Web donde estará emplazada la aplicación, una PC Servidor de Base de Datos en la cual estará la base de datos del sistema y una Impresora para imprimir los reportes que genere el sistema. Para una mayor comprensión consultar Figura 4.1.

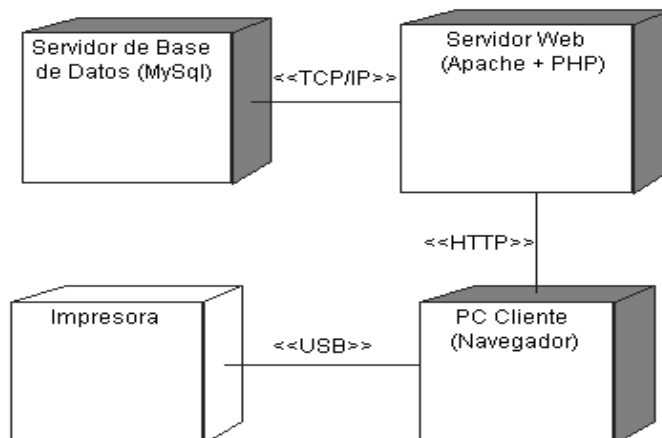


Figura 4.1 Diagrama de Despliegue.

4.3 Modelo de Implementación.

El modelo de Implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. Los diagramas de componentes se muestran en el Anexo 5 debido a su extensión y cantidad.

4.4 Modelo de Prueba.

Se ha decidido realizar el modelo de prueba dirigido a las partes o funcionalidades más críticas dentro del funcionamiento del sistema. Para ello se hizo un estudio de las posibles áreas críticas por cada caso de uso del sistema. Y una vez identificada, se realiza inicialmente una descripción textual de lo que ocurre en el mundo real, por lo que es necesario ejecutar el caso de prueba, precisando además, la información de entrada y los comandos a ejecutar por el actor. Además, se describe textualmente el estado de la información almacenada. Luego se describe textualmente el estado en el que queda la información y las alertas que puedan generarse, una vez ejecutado el caso de uso con los valores y el estado especificado en la entrada, por último se exponen las condiciones que deben cumplirse mientras se ejecuta el caso de prueba.

Tomando en consideración la cantidad de casos de uso que presenta la solución propuesta, el número de actividades contenidas en éstos y la cantidad de pruebas que se requieren realizar para cada uno, se expondrán solamente el modelo de prueba correspondiente a los casos de uso críticos del sistema y a las partes o funcionalidades más vulnerables dentro del funcionamiento del mismo.

4.4.1 Caso de Uso Gestionar Cliente.

Para el caso de uso Gestionar Cliente se ha decidido hacer las pruebas a uno de los datos a recoger a la hora de insertar un cliente en la base de datos. Se refiere específicamente a la cuenta bancaria en CUP de la cual se extrae el número de sucursal a la que pertenece la misma. Como se ve es un dato del cual depende otro, por lo que si se garantiza que éste sea introducido correctamente, se está acreditando la validez de ambos.

Prueba para una clase inválida.

Entrada	Resultados	Condiciones
<p>El caso de prueba tiene lugar cuando el administrador decide insertar un cliente nuevo en la base de datos del sistema. Para ello deberá seleccionar la opción Adicionar Cliente ubicada en el menú. El sistema por su parte mostrará una interfaz con el formulario correspondiente, listo para realizar la inserción de los datos.</p> <p>Nombre: Ministerio de Educación Siglas: MINED Teléfono: 865924 Fax:5224785 Representante Legal: Alexánder Santos De la Horra País: Cuba Objeto Social: Educación Domicilio Legal: Boyeros # 324 Número de la cuenta bancaria en CUP: 452k85 Número de la cuenta bancaria en CUC: 5824657 Nombre de Banco: Metropolitano</p> <p>Una vez que este dato sea introducido, el administrador presionará el botón Adicionar Cliente.</p>	<p>Luego de ser presionado el botón Adicionar Cliente el sistema muestra un mensaje de error informando que el número de cuenta bancaria en CUP no cumple con el formato correcto.</p>	<p>Los datos deben estar en el formato correcto. El cliente que se intenta adicionar no puede estar registrado con anterioridad.</p>

Prueba para una clase válida.

Entrada	Resultados	Condiciones
<p>El caso de prueba tiene lugar cuando el administrador decide insertar un cliente nuevo en la base de datos del sistema. Para ello deberá seleccionar la opción Adicionar Cliente ubicada en el Menú. El sistema por su parte mostrará una interfaz con el formulario correspondiente, listo para realizar la inserción de los datos.</p>	<p>Luego de ser presionado el botón Adicionar Cliente, el sistema inserta los datos en la base de datos de la siguiente forma:</p> <p>Nombre: Ministerio de Educación Siglas: MINED Teléfono: 865924 Fax:5224785 Representante Legal: Alexánder Santos De la Horra</p>	<p>Los datos deben estar en el formato correcto. El cliente que se intenta adicionar no puede estar registrado con anterioridad.</p>

<p>Nombre: Ministerio de Educación Siglas: MINED Teléfono: 865924 Fax: 5224785 Representante Legal: Alexánder Santos De la Horra País: Cuba Objeto Social: Educación Domicilio Legal: Boyeros # 324 Número de la cuenta bancaria en CUP: 452985 Número de la cuenta bancaria en CUC: 5824657 Nombre de Banco: Metropolitano</p> <p>Una vez introducidos los datos, el administrador presionará el botón Adicionar Cliente.</p>	<p>País: Cuba Objeto Social: Educación Domicilio Legal: Boyeros # 324 Número de la cuenta bancaria en CUP: 452985 Sucursal: 985 Número de la cuenta bancaria en CUC: 5824657 Nombre de Banco: Metropolitano</p> <p>Después de ser insertados muestra un mensaje de confirmación al administrador informándole que los datos del cliente fueron insertados satisfactoriamente.</p>	
---	--	--

4.4.2 Caso de Uso Administrar Proyecto.

Para el caso de uso Administrar Proyecto se ha decidido hacer las pruebas a los datos relacionados con el correo electrónico del responsable de proyecto por parte del cliente y el correo electrónico del responsable de proyecto por la parte UCI. Puesto que una dirección de correo es un dato que debe tener un formato específico y porque además se garantizaría no perder el contacto con ambos responsables.

Prueba para una clase válida.

Entrada	Resultados	Condiciones
<p>El caso de prueba tiene lugar cuando el administrador desea ingresar un nuevo Proyecto al sistema. Para realizar dicha acción tiene dos vías: Una sería, una vez insertado el Cliente el sistema preguntará si desea continuar con la inserción del Proyecto asociado a él, y la otra de manera independiente, para lo cual deberá seleccionar la opción Adicionar Proyecto ubicada en el Menú. De ambas maneras el sistema mostrará un formulario</p>	<p>Luego de ser presionado el botón Adicionar Proyecto, el sistema muestra un mensaje de error informando que el correo electrónico del responsable de proyecto por la parte del cliente no cumple con el formato correcto, e indica el formato del mismo.</p> <p>Una vez corregido el error el sistema vuelve a mostrar un mensaje de error, esta vez informando que la dirección de correo del responsable de</p>	<p>Los datos deben estar en el formato correcto.</p> <p>El proyecto que se intenta adicionar no puede estar registrado con anterioridad.</p> <p>Debe existir un cliente al cual asociar dicho proyecto.</p>

<p>listo para la inserción del Proyecto.</p> <p>Nombre del proyecto: Historia Universal Nombre el responsable por la parte Cliente: Sandy O. Rondón Cedeño Correo del responsable por la parte cliente: srondon2uci.cu Nombre del responsable por la UCI: Isabel María Calas Olazabal Correo del responsable por la UCI: imcalas.uci.cu Área de trabajo del responsable por la UCI: Facultad 10 Productos asociados: Historia Antigua, Historia de Cuba.</p> <p>Una vez introducidos los datos, el administrador presionará el botón Adicionar Proyecto.</p>	<p>proyecto por la parte UCI, no tiene el formato correcto, e indica el formato del mismo.</p>	
---	--	--

Prueba para una clase válida.

Entrada	Resultados	Condiciones
<p>El caso de prueba tiene lugar cuando el administrador desea ingresar un nuevo Proyecto al sistema. Para realizar dicha acción tiene dos vías: Una sería, una vez insertado el Cliente el sistema preguntará si desea continuar con la inserción del Proyecto asociado a él, y la otra de manera independiente, para lo cual deberá seleccionar la opción Adicionar Proyecto ubicada en el Menú. De ambas maneras el sistema mostrará un formulario listo para la inserción del Proyecto.</p> <p>Nombre del proyecto: Historia Universal Nombre el responsable por la parte Cliente: Sandy O. Rondón Cedeño</p>	<p>Luego de ser presionado el botón Adicionar Proyecto, el sistema inserta los datos en la base de datos de la siguiente forma:</p> <p>Nombre del proyecto: Historia Universal Nombre el responsable por la parte Cliente: Sandy O. Rondón Cedeño Correo del responsable por la parte cliente: srondon@uci.cu Nombre del responsable por la UCI: Isabel María Calas Olazabal Correo del responsable por la UCI: imcalas@uci.cu Área de trabajo del responsable por la UCI: Facultad 10 Productos asociados: Historia</p>	<p>Los datos deben estar en el formato correcto.</p> <p>El proyecto que se intenta adicionar no puede estar registrado con anterioridad.</p> <p>Debe existir un cliente al cual asociar dicho proyecto.</p>

<p>Correo del responsable por la parte cliente: sronдон@uci.cu Nombre del responsable por la UCI: Isabel María Calas Olazabal Correo del responsable por la UCI: imcalas@uci.cu Área de trabajo del responsable por la UCI: Facultad 10 Productos asociados: Historia Antigua, Historia de Cuba.</p> <p>Una vez introducidos los datos, el administrador presionará el botón Adicionar Proyecto.</p>	<p>Antigua, Historia de Cuba.</p> <p>Después de ser insertados muestra un mensaje de confirmación al administrador informándole que los datos del Proyecto fueron insertados satisfactoriamente.</p>	
---	--	--

4.4.3 Caso de Uso Administrar Producto.

Para el caso de uso Administrar Producto se ha decidido hacer las pruebas al dato relacionado con el archivo a guardar. Puesto que éstos para poder ser salvados en la ubicación correspondiente deben tener un formato específico para garantizar que cuando sea necesario consultar la información que contienen sea perfectamente posible hacerlo.

Prueba para una clase inválida. (Extensión inválida)

Entrada	Resultados	Condiciones
<p>El caso de prueba se inicia cuando el Líder de Proyecto o el Administrador desean ingresar un nuevo producto al sistema. Para ello, tienen dos vías, una consiste en, una vez insertado el proyecto, el sistema preguntará si desea continuar con los productos asociados, y de otra forma el usuario deberá seleccionar la opción Adicionar Producto ubicada en el menú. En ambos casos el sistema por su parte muestra un formulario listo para insertar los datos del producto: Nombre del producto: Historia Universal Tipo de producto: Multimedia educativa Está certificado o no por CALISOFT: si</p>	<p>Después de ser presionado el botón Adicionar Producto el sistema muestra un mensaje de error informando al usuario que la extensión del archivo que se intenta subir no cumple con el formato correcto.</p>	<p>Los datos deben estar en el formato correcto. El producto que se intenta adicionar no puede estar registrado con anterioridad. Debe existir un proyecto al cual asociar dicho producto.</p>



<p>Se tiene un ejemplar del original: si</p> <p>Manual de Usuario: C:\Documents and Settings\Administrador\Escritorio\manual.docx</p> <p>Una vez insertados los datos, el usuario presionará el botón Adicionar Producto.</p>		
--	--	--

Prueba para una clase inválida. (Nombre inválido)

Entrada	Resultados	Condiciones
<p>El caso de prueba se inicia cuando el Líder de Proyecto o el Administrador desean ingresar un nuevo producto al sistema. Para ello, tienen dos vías, una consiste en, una vez insertado el proyecto el sistema preguntará si desea continuar con los productos asociados, y de otra forma el usuario deberá seleccionar la opción Adicionar Producto ubicada en el menú. En ambos casos el sistema por su parte muestra un formulario listo para insertar los datos del producto:</p> <p>Nombre del producto: Historia Universal</p> <p>Tipo de producto: Multimedia educativa</p> <p>Está certificado o no por CALISOFT: si</p> <p>Se tiene un ejemplar del original: si</p> <p>Manual de Usuario: C:\Documents and Settings\Administrador\Escritorio\manu@l.doc</p> <p>Una vez insertados los datos, el usuario presionará el botón Adicionar Producto.</p>	<p>Después de ser presionado el botón Adicionar Producto el sistema muestra un mensaje de error informando al usuario que el Nombre del archivo que se intenta subir no cumple con el formato correcto.</p>	<p>Los datos deben estar en el formato correcto.</p> <p>El producto que se intenta adicionar no puede estar registrado con anterioridad.</p> <p>Debe existir un proyecto al cual asociar dicho producto.</p>

Prueba para una clase inválida. (Tamaño del archivo inválido)

Entrada	Resultados	Condiciones
<p>El caso de prueba se inicia cuando el Líder de Proyecto o el Administrador desean ingresar un nuevo producto al sistema. Para ello, tienen dos vías, una consiste en, una vez insertado el proyecto el sistema preguntará si desea continuar con los productos asociados, y de otra forma el usuario deberá seleccionar la opción Adicionar Producto ubicada en el menú. En ambos casos el sistema por su parte muestra un formulario listo para insertar los datos del producto:</p> <p>Nombre del producto: Historia Universal Tipo de producto: Multimedia educativa Está certificado o no por CALISOFT: si Se tiene un ejemplar del original: si</p> <p>Manual de Usuario: C:\Documents and Settings\Administrador\Escritorio\manual.doc (Tamaño mayor que 5 Mb)</p> <p>Una vez insertados los datos, el usuario presionará el botón Adicionar Producto.</p>	<p>Después de ser presionado el botón Adicionar Producto el sistema muestra un mensaje de error informando al usuario que el tamaño del archivo que se intenta subir excede el límite aceptado.</p>	<p>Los datos deben estar en el formato correcto. El producto que se intenta adicionar no puede estar registrado con anterioridad. Debe existir un proyecto al cual asociar dicho producto.</p>

Prueba para una clase válida.

Entrada	Resultados	Condiciones
<p>El caso de prueba se inicia cuando el Líder de Proyecto o el Administrador desean ingresar un nuevo producto al sistema. Para ello, tienen dos vías, una consiste en, una vez insertado el proyecto el sistema preguntará si desea continuar con los productos asociados, y de otra forma el usuario deberá seleccionar la opción Adicionar Producto</p>	<p>Después de ser presionado el botón Adicionar Producto el sistema inserta los datos del producto en la base de datos de la siguiente manera:</p> <p>Nombre del producto: Historia Universal Tipo de producto: Multimedia educativa Está certificado o no por CALISOFT: si</p>	<p>Los datos deben estar en el formato correcto. El producto que se intenta adicionar no puede estar registrado con anterioridad. Debe existir un proyecto al cual asociar dicho</p>

<p>ubicada en el menú. En ambos casos el sistema por su parte muestra un formulario listo para insertar los datos del producto: Nombre del producto: Historia Universal Tipo de producto: Multimedia educativa Está certificado o no por CALISOFT: si Se tiene un ejemplar del original: si Manual de Usuario: C:\Documents and Settings\Administrador\Escritorio\manual.doc Una vez insertados los datos, el usuario presionará el botón Adicionar Producto.</p>	<p>Se tiene un ejemplar del original: si El documento adjuntado quedará salvado en la carpeta correspondiente a ese producto dentro del sitio.</p>	<p>producto.</p>
---	--	------------------

4.4.4 Caso de Uso Gestionar Contrato.

Para el caso de uso Gestionar Contrato se ha decidido hacer las pruebas al tema relacionado con la forma de pago que se empleará para pagar el producto en cuestión. En caso de que la forma de pago sea a través de por ciento o por anticipo, se debe introducir un grupo de acciones con el monto a pagar y el por ciento que representa respectivamente. El objetivo sería entonces verificar que el por ciento insertado sea un número menor que 100.

Prueba para una clase inválida.

Entrada	Resultados	Condiciones
<p>El caso de prueba tiene lugar cuando el Administrador desea adicionar un nuevo contrato en el sistema. Para ello deberá seleccionar la opción Adicionar Contrato, ubicada en el menú del sistema. El sistema por su parte mostrará un formulario listo para insertar los datos del Contrato. Número del contrato: HA1 Tipo de contrato: Guionista Descripción del objeto contrato: Solicitar la realización del guión de la multimedia de</p>	<p>Después de ser presionado el botón Adicionar Contrato el sistema muestra un mensaje de error informando al usuario que el por ciento que representa la acción a pagar no cumple con el formato correcto debido a que el por ciento debe ser un número menor que 100.</p>	<p>Debe existir un cliente en el sistema con su respectivo proyecto al cual asociarle dicho contrato. El contrato que se desea adicionar no debe existir en la base de datos.</p>



<p>Historia Antigua. Monto del contrato: 300 \$ Tipo de Moneda: MN Vigencia del contrato: 1 año Forma de Pago: Por por ciento Tipo de Transferencia: cheque Acción: Mayas Monto: 100 \$ Por ciento: 106% Luego de insertar los datos de contrato, el administrador presionará el botón Adicionar Contrato.</p>		
---	--	--

Prueba para una clase válida.

Entrada	Resultados	Condiciones
<p>El caso de prueba tiene lugar cuando el Administrador desea adicionar un nuevo contrato en el sistema. Para ello deberá seleccionar la opción Adicionar Contrato, ubicada en el menú del sistema. El sistema por su parte mostrará un formulario listo para insertar los datos del Contrato. Número del contrato: HA1 Tipo de contrato: Guionista Descripción del objeto contrato: Solicitar la realización del guión de la multimedia de Historia Antigua. Monto del contrato: 300 \$ Tipo de Moneda: MN Vigencia del contrato: 1 año Forma de Pago: Por por ciento Tipo de Transferencia: cheque Acción: Mayas Monto: 100 \$ Por ciento: 20% Luego de insertar los datos de contrato, el administrador presionará el botón Adicionar Contrato.</p>	<p>Después de ser presionado el botón Adicionar Contrato el sistema inserta los datos del contrato en la base de datos de la siguiente forma: Número del contrato: HA1 Tipo de contrato: Guionista Descripción del objeto contrato: Solicitar la realización del guión de la multimedia de Historia Antigua. Monto del contrato: 300 \$ Tipo de Moneda: MN Vigencia del contrato: 1 año Forma de Pago: Por por ciento Tipo de Transferencia: cheque Acción: Mayas Monto: 100 \$ Por ciento: 20% Luego muestra un mensaje informándole al administrador que los datos del contrato fueron insertados correctamente.</p>	<p>Debe existir un cliente en el sistema con su respectivo proyecto al cual asociarle dicho contrato. El contrato que se desea adicionar no debe existir en la base de datos.</p>

Como se ha podido apreciar se han realizado las pruebas a las zonas vulnerables para cada caso de uso crítico dentro del sistema. Pudiéndose comprobar que el sistema cumple con las características idóneas para su puesta en funcionamiento. Se hace esta afirmación sobre la base, de que si las pruebas realizadas resultaron satisfactorias en las áreas seleccionadas, entonces las demás funcionalidades del sistema cumplirán los requisitos deseados.

4.5 Conclusiones del Capítulo.

En este capítulo se ha dejado en evidencia mediante el diagrama de despliegue, los nodos necesarios para la utilización del sistema, lo que se traduce en la utilización de una PC servidora para emplazar la aplicación, una PC servidora de base de datos para la BD, las PC clientes desde donde el usuario podrá interactuar con la aplicación y una impresora para imprimir los reportes que el sistema genere. Además se mostró la relación existente entre los archivos de la aplicación, mediante los diagramas de componentes. Y por último se comprueba el correcto funcionamiento del sistema mediante las pruebas de integración.

CONCLUSIONES

- La situación actual de protección legal en el desarrollo de Software Educativo que se manifiesta en la universidad, confirman la necesidad del desarrollo de un sistema informático que automatice las actividades que se requieren realizar para gestionar, organizar y controlar toda la información y la documentación técnico - legal requerida para la protección de los productos de Software Educativo.
- Se realizó un estudio del marco teórico - conceptual asociado al dominio del problema, permitiendo este análisis seleccionar la metodología que apoya la solución del problema resultando la más idónea para cumplimentar el proceso de desarrollo las definiciones de RUP y se determinaron las herramientas de desarrollo a emplear en la producción del software donde se propuso la utilización del lenguaje PHP, con su motor de plantillas Smarty y su capa de acceso a datos ADOdb, el gestor de bases de datos MySql y el servidor Web Apache.
- Se realizó el análisis y diseño de la propuesta del sistema lográndose identificar las funcionalidades y la forma en que estas deben ser implementadas.
- Se desarrolló el sistema automatizado logrando abarcar en él todas las actividades que contribuyen a la correcta protección legal que se le debe conferir a los productos de Software Educativo, obteniéndose de esta forma una herramienta que facilita, viabiliza y apoya este proceso.

RECOMENDACIONES

Se recomienda desarrollar nuevos módulos encaminados a la gestión de la información legal de otros tipos de software que se desarrollen en la Universidad.

Extender el uso de la aplicación a otras instituciones del país que desarrollen este tipo de productos.

Archivar el documento con el objetivo de que si en un futuro fuera necesario reparar el software o agregarle nuevos módulos, dichas tareas puedan llevarse a cabo de manera eficiente y sencilla.

REFERENCIAS

1. ANAYA, R. *El proceso de desarrollo de software: una visión desde la academia y la industria*. Universidad EAFIT, Grupo de Investigación en Ingeniería de Software, 2005. 46 p.
2. BÁEZ, C. D. *Estrategia Técnico Legal para la protección de Software Educativo en la Universidad de las Ciencias Informáticas*. XVI Forum de Ciencia y Técnica, Segunda Etapa. 2006.
3. CANSECO, R. E. *Metodología MSF*, 2007. Disponible en: <http://www.coworker.com.mx/nota.asp?id=28> . Consultado el (15/12/2006).
4. CEYUSA, D. *El Patrón de acceso a datos (DAO)*, 2007. Disponible en: <http://www.glib.org.mx/article.php?story=20060702185110418> Consultado el (10/3/2007).
5. FERNÁNDEZ, J. *El Patrón Fachada*, Departamento de Informática y Automática, Universidad de Salamanca, 2007. Disponible en: <http://zarza.fis.usal.es/~fgarcia/docencia/poo/03-04/Trabajos/Fachada.pdf> Consultado el (12/3/2007).
6. HIETANIEMI, J. *Perl*, 2007d. Disponible en: <http://es.wikipedia.org/wiki/Perl> Consultado el (17/2/2007).
7. JACOBSON, I. *El proceso Unificado de desarrollo de software*. Edición en español, Madrid 2000. ISBN 84-7829-036-2. 2007.
8. TORBEN, W. L. *Programación Web*. Disponible en: <http://lenguajes-de-programacion.com/programacion-web.shtml> Consultado el (17/2/2007).
9. LARMAN, C. *UML y Patrones: Introducción al análisis y diseño orientado a objetos y al proceso unificado*, 2002. Segunda Edición por Prentice Hall.
10. MARTÍNEZ, P. Y. *Flujo de Trabajo del Proceso Productivo del Software Educativo en la UCI*. Universidad de las Ciencias Informáticas. XVI Forum de Ciencia y Técnica, Segunda Etapa. 2006.
11. MICROSOFT FOUNDATION. *Descripción general de Microsoft SQL Server 2005*. 2007. Disponible en: <http://www.microsoft.com/latam/sql/2005/productinfo/> Consultado el (17/2/2007).

12. MENDOZA, S. M. *Metodologías De Desarrollo De Software*, 2007. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html Consultado el (15/12/2006).
13. MASIP D. *¿Qué es Oracle?*, 2007. Disponible en: <http://www.desarrolloweb.com/articulos/840.php> Consultado el (22/2/2007).
14. SCHUMACHER, R. *MySQL*, 2007. Disponible en: <http://es.wikipedia.org/wiki/MySQL> Consultado el (22/2/2007).
15. VALLE, J. G. *Definición arquitectura cliente servidor*, 2007. [2007]. Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml> Consultado el (15/12/2006).
16. VÁZQUEZ, J. *Desarrollo Web con PHP y MYSQL*, 2007. Ediciones Anaya Multimedia. Disponible en: <http://www.gnuinos.com/drupal/?q=node/84> Consultado el (17/2/2007).
17. WALTER, S. *Nuevas características de seguridad en ASP.NET 2.0*. 2006 Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/100904/voices/SecFeatNT2.asp> Consultado el (17/2/2007).
18. WIKIMEDIA FOUNDATION, I. *Sistemas de Gestión de Bases de Datos*, 2007. Disponible en: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos Consultado el (22/2/2007).

BIBLIOGRAFÍA

1. GILFILLAN, I. *La Biblia de MySql*, 2003. Ediciones Anaya Multimedia. 889 p.
2. GALLEGO, V. *Desarrollo Web con PHP y MySql*, 2003. Ediciones Anaya Multimedia. 288 p.
3. ZELDMAN, J. *Diseño con estándares Web*, 2003. Ediciones Anaya Multimedia. 400 p.
4. MUSIANO, C. *HTML: La guía completa*, 2005. Ediciones O' Reilly. 546 p.
5. FRENTZEN, J. *Superutilidades para Javascript*, 2003. McGRAW-HILL/Interamericana de España, S.A.U.
6. HITSCHFELD, K. N. *Tutorial de UML*, 2005. Departamento de Ciencias de la Computación, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile. Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/>. Consultado el (17/12/2006)
7. ZMIEVSKI, A. *Manual de Smarty*, 2005. 174 p.
8. SAETHER, B. S. *Manual PHP*, 2001. 258 p.
9. LIM, J. *Manual ADOdb*, 2006. 296 p.

GLOSARIO DE TÉRMINOS Y SIGLAS

BD: sigla que se utiliza para referirse al término Base de Datos.

CENDA: es el Centro Nacional del Derecho de Autor de nuestro país.

CUS: sigla que se utiliza para referirse al término Caso de Uso del Sistema.

HTTP: Es el conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

HTML: Lenguaje usado para escribir documentos para servidores World Wide Web.

Integer: tipo de dato que incluye los números naturales, sus opuestos y el cero.

Logueado: estado en que se encuentra un usuario del sistema una vez que introduce su usuario y contraseña, y el sistema le permite realizar determinadas acciones.

MD5: Algoritmo para encriptar las contraseñas de los usuarios.

String: en informática es traducido como cadena de caracteres de longitud variable pueden oscilar entre 0 y aproximadamente 2 mil millones de caracteres.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

USB: Interfaz estándar que facilita la conexión de periféricos a un ordenador.

Web: Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la Web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

Encriptar: Acción en la cual a través de una función matemática se sustituyen los caracteres originales de la contraseña por otros, con el objetivo de brindar una seguridad adecuada.