



Universidad de las Ciencias Informáticas

***Análisis y Diseño del módulo Exclusión Mutua y Sincronización
del subsistema de autoaprendizaje para el Laboratorio
Virtual de Sistemas Operativos.***

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

Autora: Suset Pérez Basulto.

Tutor: Ing. Carlos Y. Hidalgo García.

Ciudad de la Habana, Cuba

Mayo, 2011

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro ser la única autora de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas a hacer uso exclusivo de del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de Mayo del año 2011.

Firma del autor

Suset Pérez Basulto

Firma del tutor

Carlos Yasmany Hidalgo García

Agradecimientos

Primeramente y de manera muy especial agradecerles a mis padres por haber sido siempre incondicionales conmigo, por apoyarme siempre en los momentos difíciles de mi vida y darme fuerzas cuando me encontraba angustiada, por haber confiado siempre en mi, brindándome cada día a pesar de la distancia que nos separaba el inmenso amor, el más grande amor que existe que es el de los padres e hijos.

A mi hermana querida Suliet por haberme comprendido siempre a pesar de su poca experiencia.

A mi tutor Carlos por haberme apoyado y guiado en todo el desarrollo de la tesis.

A mi novio Alberto, gracias mi ti por la compañía, amor y comprensión que me has brindado durante estos 5 años.

A mi familia, mis abuelos, mis tías en especial a mi tía Odalis por ser tan atenta y cariñosa conmigo, a mis primos por ayudarme siempre.

A mis suegros que tanto apoyo me han brindado.

A mis amistades Yanelis, Yanet, Odaisa y mis compañeros de aula.

A la profesora Zoraida por haber sido como una madre para mí.

A todos Muchas Gracias.



Dedicatoria

Le dedico el Trabajo de Diploma a mis padres por ser lo más grande y maravilloso que tengo en la vida.

A mi madre por ser mi amiga y compañera, consejera y aliada, te amo mamita linda.

A mi padre por ser para mí un ejemplo de hombre digno, y el mejor papa del mundo.

A mi hermana Suliet, gracias tata por tu cariño.

A mi novio Alberto por disfrutar de mis alegrías y padecer de mis sufrimientos, Gracias amorcito.

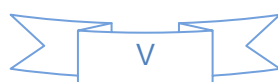
Resumen

Con el surgimiento del Modelo Integral de Formación – Producción – Investigación en la UCI, surge la necesidad de perfeccionar el plan de estudios del Ingeniero en Ciencias Informáticas con el fin de elevar la calidad del egresado. Como parte de este perfeccionamiento se decidió desarrollar un Laboratorio Virtual de Sistemas Operativos ya que la asignatura no se encontraba preparada para incorporar los cambios necesarios, donde los estudiantes mediante un entorno virtual de aprendizaje reciben sus clases de forma semipresencial. Para llevar a cabo este proceso se necesitaba de la existencia de un módulo de autoaprendizaje donde el estudiante pudiera adquirir todo el conocimiento teórico necesario de los temas de la asignatura.

En este trabajo se realizó un estudio de los diferentes laboratorios virtuales y los sistemas de autoaprendizaje existentes, así como las herramientas y lenguajes de modelado, definiendo un marco de trabajo que permite confeccionar el diseño del módulo de Exclusión Mutua y Sincronización del subsistema de autoaprendizaje para el Laboratorio Virtual de Sistemas Operativos, obteniendo todos los artefactos generados en el diseño.

Para la validación de los requisitos funcionales y los casos de usos descritos se aplicaron Métricas de calidad y una Matriz de trazabilidad, donde se obtuvieron resultados satisfactorios.

Palabras Claves: Aplicaciones Educativas, Laboratorio Virtual, Sistemas de Autoaprendizaje.



Índice de Contenido

Contenido

Introducción	12
1.1 Sistemas de Autoaprendizaje	16
1.2 Laboratorios Virtuales	16
1.3 Diseño de Aplicaciones Multimedia	17
1.4 Recursos Didácticos y Objetos de Aprendizaje	19
1.5 Ingeniería de Requisitos	19
1.6.1 Extracción de Requisitos.....	20
1.6.2 Análisis de Requisitos	21
1.6.3 Especificación de Requisitos.....	21
1.6.4 Validación de Requisitos	22
1.6.5 Gestión de Requisitos	23
1.6 Metodologías de desarrollo de Software	24
1.6.1 Rational Unified Pocess (RUP)	24
1.6.2 Exreme Programining (XP)	24
1.6.3 Relationship Management Methodology (RMM)	24
1.6.4 Marco de trabajo	25
1.7 Herramientas CASE	27
1.7.1 Rational Rose Enterprise Edition.....	27
1.7.2 Enterprise Architect.....	28
1.7.3 Visual Paradigm.....	28
1.8 Lenguaje de Modelado	28
1.8.1 Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia (OMMMA – L).....	28

Índice de Contenido

1.8.2 El Lenguaje de Modelado Unificado (UML)	29
1.8.3 Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)	29
1.9 Patrones	30
1.9.1 Patrones de Casos de Uso	30
1.9.2 Patrón Arquitectónico	31
1.9.3 Patrones de Diseño	32
1.13 Conclusiones del Capítulo	33
Capítulo 2 : Descripción de la solución	35
2.1 Introducción	35
2.2 Mapa Conceptual	35
2.2 Requisitos funcionales y no funcionales	36
2.3 Diagrama de Casos de Uso	38
2.4 Descripción Textual	39
2.5 Diagrama de Actividades	43
2.5 Diagrama de Navegación	45
2.6 Diagrama de Presentación	48
2.7 Diagrama de Secuencia	49
2.8 Conclusiones del capítulo	49
Capítulo 3 Validación	51
3.1 Introducción	51
3.2 Validación de Requisitos mediante Métricas de la calidad para la especificación.	51
3.3 Modelo de Métricas OO aplicadas al Diagrama de Casos de Uso del Sistema.	52
3.5 Matriz de trazabilidad	61
3.6 Validación del Diseño	62

Índice de Contenido

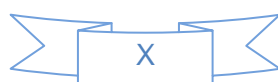
3.6.1 Métrica Tamaño operacional de clase (TOC)	62
3.6.2 Métrica de Relaciones entre clases (RC)	67
3.7 Conclusiones del capítulo.....	74
Conclusiones	75
Recomendaciones	76
Referencias Bibliográficas.....	77
Bibliografía	79

Índice de Figuras

Figura 1: Mapa Conceptual	35
Figura 2: Diagrama de Caso de Uso	39
Figura 3: Diagrama de Actividades del caso de uso Habilitar Contenido.....	43
Figura 4: Diagrama de Clases del caso de uso Habilitar Contenido.	45
Figura 5: Diagrama de Navegación del caso de uso Habilitar Contenido.....	47
Figura 6: Diagrama de Presentación del caso de uso Habilitar Contenido.	48
Figura 7: Diagrama de Secuencia del caso de uso Habilitar Contenido.	49
Figura 8: Resultados de la Métrica.	60
Figura 9: Gráfica de los resultados de la evaluación de la métrica TOC agrupados por las tendencias de los Valores.	64
Figura 10: Representación en % de los resultados obtenidos en el instrumento agrupados en intervalos definidos.	65
Figura 11: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.	65
Figura 12: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad.	66
Figura 13: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.....	67
Figura 14: Gráfica de los resultados de la evaluación de la Métrica RC agrupados por las tendencias de los valores.	69
Figura 15: Representación en % de los resultados obtenidos en el instrumento agrupados en intervalos definidos.	70
Figura 16: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Acoplamiento.	71
Figura 17: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Mantenimiento.....	72
Figura 18: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Mantenimiento.....	73

Índice de Figuras

Figura 19: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Reutilización.....74



Índice de Tablas

Tabla 1: Acciones por las que están compuestas las actividades genéricas.	27
Tabla 2: Descripción textual del caso de uso Habilitar Contenido.....	42
Tabla 3: Métricas para validar Diagrama de Caso de Uso del Sistema.	56
Tabla 4: Métricas para validar Diagrama de Caso de Uso del Sistema.	59
Tabla 5: Matriz de trazabilidad de requisitos	62
Tabla 6: Cantidad de clases y promedio de procedimientos.....	63
Tabla 7: Cantidad de procedimientos por clases.	64
Tabla 8: Atributo Responsabilidad.	65
Tabla 9: Atributo Complejidad.	65
Tabla 10: Atributo Reutilización.....	66
Tabla 11: Cantidad de clases y promedio de asociaciones de uso.....	68
Tabla 12: Cantidad de dependencias por clasificación.....	69
Tabla 13: Atributo Acoplamiento.	70
Tabla 14: Atributo Complejidad de Mantenimiento.	71
Tabla 15: Atributo Cantidad de pruebas.....	72
Tabla 16: Atributo Reutilización.....	73

Introducción

Introducción

En el mundo contemporáneo el uso de las Tecnologías de la Información y las Comunicaciones (TIC), constituye una necesidad para el desarrollo de cualquier país. Es un gran reto que requiere de la formación de una cultura informática en las nuevas generaciones.

En Cuba se está llevando a cabo gradualmente la informatización de varios sectores, con el uso de las nuevas soluciones informáticas junto a las TIC, para el desarrollo de sistemas dirigidos a automatizar procesos de gran importancia para la sociedad. Uno de los sectores que se ha beneficiado con estos cambios es sin duda el sector Educativo donde se han creado diferentes medios didácticos como los entornos virtuales de aprendizaje (EVA), este medio le sirve de gran ayuda a los profesores y estudiantes al Proceso de Enseñanza-Aprendizaje de forma online.

Entre los diferentes programas que ha llevado la Revolución Cubana para el desarrollo de esta importante labor se encuentra la creación de la Universidad de las Ciencias Informáticas (UCI) en el año 2002 donde cada curso se forman miles de ingenieros de diferentes lugares del país y su principal tarea es desarrollar la industria del software cubano. Para lograr este propósito se ha adoptado un modelo de enseñanza y formación, donde los estudiantes pueden recibir sus clases de forma semipresencial de las distintas asignaturas para de esta forma dedicarle más tiempo a la producción e investigación, de este modo el estudiante se convierte en el principal responsable de su autoaprendizaje.

Lamentablemente no todas las asignaturas de la carrera cuentan con estos medios didácticos y una de ellas es Sistemas Operativos la cual tiene un fuerte contenido teórico. Esta asignatura tiene muy poca interacción con las TIC, esto trae como consecuencia que necesita de muchas actividades presenciales, entrando en contradicción con el Modelo Integral de Formación – Producción – Investigación.

Para resolver este problema se ha creado un Laboratorio Virtual de Sistemas Operativos que será incorporado a la Plataforma Moodle. El Laboratorio Virtual es un proyecto compuesto por 6 subsistemas y 17 módulos, donde los estudiantes recibirán todo el contenido de la asignatura. Las aplicaciones desarrolladas como parte del Laboratorio Virtual de Sistemas Operativos no cuentan con los módulos de autoaprendizaje, para permitir que los estudiantes adquieran los conocimientos necesarios desde el punto de vista teórico.

Introducción

Para la informatización de este proceso, es de suma importancia realizar un análisis previo identificando las necesidades del cliente, llevándolo a un lenguaje fácil de entender por los desarrolladores, ya que este aspecto es de suma importancia pues a veces resulta muy complejo esta comunicación entre analistas e implementadores.

Con lo anteriormente planteado se define el **problema de investigación**: ¿Cómo lograr un entendimiento entre clientes y desarrolladores sobre el Tema Exclusión Mutua y Sincronización para el subsistema de autoaprendizaje del LV de SO, que permita la posterior implementación del mismo?

Definiendo como **Objeto de estudio**: Procesos de Desarrollo de Software y **Campo de acción**: Análisis y Diseño dentro del Proceso de Desarrollo de Software para aplicaciones de enseñanzas multimedia.

Como **Objetivo General**: Realizar el Análisis y Diseño del módulo Exclusión Mutua y Sincronización del subsistema de autoaprendizaje para el Laboratorio Virtual de Sistemas Operativos, que permita el desarrollo posterior de dicho módulo.

Objetivos Específicos:

- Elaborar el marco teórico de la investigación.
- Obtener los artefactos generados en el Análisis y Diseño.
- Validar los artefactos obtenidos.

Idea a defender: Realizando el Análisis y Diseño del módulo Exclusión Mutua y Sincronización del Subsistema de Autoaprendizaje para el Laboratorio Virtual de Sistemas Operativos se logrará implementar el subsistema de autoaprendizaje.

De este modo quedaron plasmadas las siguientes **tareas**:

- Realización de un estudio de los sistemas de autoaprendizaje, Laboratorios Virtuales, recursos didácticos, objetos de aprendizaje y mapa conceptual.

Introducción

- Realización de búsquedas bibliográficas encaminadas a investigar las metodologías de desarrollo de software, herramientas CASE, herramientas para modelar el mapa conceptual, lenguajes de modelado y patrones más utilizados.
- Identificación de los aspectos fundamentales del Tema Bloqueo que se lleva a cabo en la asignatura Sistemas Operativos.
- Confección del mapa conceptual, especificación de requisitos de software, modelo del sistema, matriz de trazabilidad de los requisitos, diagrama de clases, diagrama de estructura de navegación, diagrama de estructura de presentación y diagrama de secuencia.
- Validación de los principales resultados obtenidos a través de métricas para el desarrollo de un software.

Métodos Teóricos:

- Análisis-síntesis: Se empleó para el estudio y recopilación de los conceptos referentes a la investigación, analizando documentos para la extracción de los elementos más importantes.
- Análisis histórico-lógico: Para conocer con mayor profundidad las tendencias y antecedentes más recientes de los laboratorios virtuales y de las aplicaciones multimedia, conociendo así la trayectoria histórica que tiene a través de su origen.

Métodos Empírico:

- Entrevista: Para la recopilación de información especializada o dirigida a directivos, profesores y alumnos que interactúan con las TIC en el proceso de enseñanza – aprendizaje presencial o mixto de la asignatura de Sistemas Operativos I en la Facultad 3 de la Universidad de las Ciencias Informáticas.

Para presentar los resultados obtenidos durante la investigación el trabajo consta de 3 capítulos estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica: Se describe el estado del arte referente a los Laboratorios Virtuales y los diferentes tipos que existen, también se aborda sobre las aplicaciones multimedia, dejando plasmado los principales conceptos y definiciones existentes asociados al tema, así como el lenguajes de modelado

Introducción

y herramientas a utilizar, además de realizar un estudio de las técnicas de Extracción, Especificación y Validación de Requisitos que podrían ser usadas, así como la definición de un Marco de trabajo con las actividades que se deben realizar para la obtención de los artefactos.

Capítulo 2: Se describe la solución propuesta a partir del la confección del mapa conceptual, obtención de requisitos funcionales y no funcionales, además de la modelación del diagrama de casos de uso del sistema, la descripción textual de los casos de uso y una serie de diagramas diseñados para el desarrollo del Laboratorio Virtual.

Capítulo 3: En este capítulo se validará la solución mediante las Métricas de la calidad de Especificación de Requisitos y las Métricas para validar los casos de uso del sistema, así como la realización de una Matriz de trazabilidad. Para validar el diseño se aplicaron las Métricas Tamaño Operacional de Clase y Relación entre Clases.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica

Introducción

El presente capítulo contiene el estudio de sistemas de autoaprendizaje. Para su alcance se estudia el diseño para aplicaciones educativas multimedia, consultando conceptos de multimedia para su interpretación. Haciendo énfasis en la herramienta utilizada para la estructura del Mapa conceptual el cual se convierte en el basamento teórico. Posteriormente se profundiza en las etapas de la Ingeniería de Requisitos para seleccionar las técnicas correctas para la elaboración del sistema. Se analiza las metodologías, los lenguajes de modelación, las herramientas CASE y patrones existentes, con el fin de justificar las variantes empleadas en la investigación.

1.1 Sistemas de Autoaprendizaje

El Autoaprendizaje es esencialmente importante para los estudiantes autónomos, es decir, aquellos que no toman clases en el aula con un maestro, y se refiere al proceso al que se somete un individuo, con el interés de aprender algo teórico y/o práctico. Para realizar todas estas actividades tiene que disponer de al menos un libro electrónico para el estudio de los contenidos, un Laboratorio Virtual que permite que el estudiante experimente de una manera real o simulada todo el contenido dependiendo de los objetivos y características que tenga este Laboratorio Virtual, podrá tener un sistema de autoevaluación.

El Proyecto de Laboratorio Virtual de la facultad 3 cuenta con un libro electrónico, funciona como Laboratorio Virtual además uno de los módulos por los que está compuesto es Evaluación y Ejercicios. De esta forma el mismo queda compuesto por 3 elementos fundamentales de un Sistema de Autoaprendizaje.

1.2 Laboratorios Virtuales

Los programas de Laboratorios Virtuales permiten crear un enfoque constructivo del aprendizaje donde los alumnos podrán contrastar sus hipótesis a través de una experiencia virtual. Para ello se debe de organizar una serie de actividades (prácticas virtuales) que hagan que los alumnos reflexionen continuamente acerca de la información recibida.

Capítulo 1: Fundamentación Teórica

Existen diferentes tipos de Laboratorios Virtuales como los que se muestran a continuación:

Laboratorios Virtuales de Software

Los Laboratorios Virtuales de Software, son laboratorios desarrollados como un programa de software independiente destinado a ejecutarse en la máquina del usuario, y cuyo servicio no requiere de un servidor Web. Es el caso de programas con instalación propia, que pueden estar destinados a plataformas como Linux, Windows e incluso necesitar que otros componentes de software estén instalados previamente.

Laboratorios Virtuales Remotos

Los Laboratorios Remotos son aquellos que permiten operar remotamente cierto equipamiento, en general, estos requieren de equipos servidores específicos que les den acceso a las máquinas a operar de forma remota, y no pueden ofrecer su funcionalidad ejecutándose de forma local.

Laboratorios Virtuales Web

Este tipo de laboratorio se basa en un software que depende de los recursos de un servidor determinado. Estos recursos pueden ser bases de datos, software que requieren ejecutarse en su servidor o la exigencia de determinado hardware para ejecutarse. No son programas que un usuario pueda descargar en su equipo para ejecutar localmente de forma independiente.

El Laboratorio Virtual de Sistemas Operativos es de tipo Web, el mismo brinda a los alumnos la posibilidad de aprender por cuenta propia fomentando la capacidad de análisis, el pensamiento crítico y la utilización de las tecnologías. Se convierte en una ayuda interactiva para el aprendizaje de contenidos que en la realidad son difíciles de demostrar. Es necesario acceder a un servidor para poder estudiar los contenidos y no permitir que los mismos sean descargados en la máquina del usuario.

1.3 Diseño de Aplicaciones Multimedia

Para el desarrollo de cualquier proceso se necesita realizar la modelación del mismo. El proceso didáctico que se desea llevar a cabo para que los estudiantes reciban contenidos importantes; permite reutilizar

Capítulo 1: Fundamentación Teórica

objetos de aprendizaje u otro tipo de recurso educativo. El proceso de diseño tiene dos fases esenciales que se encuentran entrelazadas.

- Fase pedagógica: Diseño de la unidad didáctica.
- Fase tecnológica: Se estandariza la unidad como OA SCORM.

Para dar cumplimiento a la primera fase se modela el dominio mediante un Mapa conceptual considerada como una herramienta didáctica de creación del aprendizaje ejecutándose de forma secuencial mediante instrucciones.(4)

Los mapas conceptuales son artefactos para la organización y representación del conocimiento. Su objetivo es representar relaciones entre conceptos en forma de proposiciones. Se estructuran en forma jerárquica en la que los conceptos más generales están en la raíz del árbol y a medida que se desciende por el mismo se van encontrando con conceptos más específicos.

Para realizar el esquema es necesario de la utilización de una herramienta capaz de representar estas relaciones de la manera más cómoda y entendible posible. Dentro de las más destacadas están:

FreeMind: Es una herramienta que permite la elaboración de mapas mentales o conceptuales. Es útil en el análisis y recopilación de información o ideas generadas en grupos de trabajo. Es la alternativa libre a la aplicación MindManager fabricada por la empresa MindJet. Su principal limitación es que se centra en un único tipo de diagramas, los mapas mentales. (5)

CmapTools: Se trata de una herramienta gratuita para entidades educativas sin ánimo de lucro que permita la construcción de mapas conceptuales. Desarrollada por el Institute for Human and Machine Cognition de la University of West Florida. Su orientación era la captura y representación del conocimiento.

El cliente estableció el uso de la herramienta **Compendium LD:** Flexible para la creación de diseños de aprendizajes. Ofrece a los usuarios la oportunidad de grabar y comunicar sus ideas de diseño con un alto nivel de detalle. Se seleccionó esta herramienta, ya que proporciona una serie de iconos para crear

Capítulo 1: Fundamentación Teórica

mapas que describen los temas y contenidos por los que está compuesto el módulo Exclusión Mutua y Sincronización. La misma es de gran ayuda para maestros y diseñadores. (6)

1.4 Recursos Didácticos y Objetos de Aprendizaje

Un recurso didáctico es todo aquel instrumento que ayuda a los profesores en su tarea de enseñar y por otra parte le facilita al estudiante el logro de los objetivos de aprendizaje. Se consideran como herramienta de ayuda para llevar a cabo la tarea formativa, siempre que se haga un uso correcto de ellos.

En la Universidad se utiliza como recurso didáctico el Entorno Virtual de Aprendizaje (EVA), siendo este, el medio tecnológico para poner en marcha el modelo Integral de Formación – Producción – Investigación. Con este recurso no es necesario para llevar a cabo el proceso de aprendizaje de manera online, se requiere además de Objetos de Aprendizajes que estandarice contenidos digitales de tal forma que sea posible la reutilización en diversos contextos educativos.

Estos Objetos de Aprendizajes deben cumplir las características:

- Formato Digital.
- Propósito pedagógico.
- Contenido interactivo. (7)

1.5 Ingeniería de Requisitos

La ingeniería de Requisitos permite gestionar las necesidades del proyecto en forma estructurada. La misma consiste en una serie de pasos organizados y bien definidos. También mejora la capacidad de predecir cronogramas de proyecto, así como sus resultados. Otro elemento importante de realizar un proceso bien detallado de IR es que disminuye los costos y retrasos del proyecto, influyendo esto en que mejore la calidad del software que debe cumplir con un conjunto de requisitos como son la funcionalidad, facilidad de uso, confiabilidad y desempeño. Además de proporcionar un consenso entre clientes y desarrolladores, que es vital para que el proyecto sea exitoso. Finalmente evita rechazos por parte de los usuarios finales ya que obliga al cliente a considerar sus requisitos cuidadosamente y revisarlos dentro del marco del problema a resolver, por lo que se le involucra durante todo el desarrollo del proyecto. (2)

Según Pressman la Ingeniería de Requisitos define 5 etapas estas son:

Capítulo 1: Fundamentación Teórica

- Extracción de Requisitos.
- Análisis de Requisitos.
- Especificación de Requisitos.
- Validación de Requisitos.
- Gestión de Requisitos.

1.6.1 Extracción de Requisitos

Esta actividad se realiza con el fin de conocer el dominio e identificar las necesidades del cliente y usuarios. Es de gran importancia que la Extracción de Requisitos sea efectiva, la misma permite la aceptación del sistema por los usuarios finales.

Existen diferentes técnicas para la Extracción de Requisitos dentro de las más usadas están:

Cuestionarios y lista de verificación: esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario. Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

Entrevistas: Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Las entrevistas pueden ser de dos tipos:

Entrevistas cerradas: Donde los Stakeholders responden a un conjunto predefinido de preguntas.

Entrevistas abiertas: Donde no hay un programa predefinido. El equipo examina una serie de cuestiones con los Stakeholders del sistema y, por lo tanto, desarrolla una mejor comprensión de sus necesidades. (8)

Tormenta de ideas: Es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas). Una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador.

Capítulo 1: Fundamentación Teórica

Como técnica de captura de requisitos es sencilla de usar y de aplicar. Además, suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del sistema, por lo que suele aplicarse en los primeros encuentros.

Sketches y Storyboards (interfaces y estructura de navegación): La misma consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard). (2)

1.6.2 Análisis de Requisitos

En esta etapa se recomienda que se realice un análisis íntegro de los requisitos teniendo en cuenta que aquí es donde se conceptúan, investigan y resaltan los problemas que puedan presentar, se buscan soluciones. Es conveniente realizar reuniones entre los desarrolladores, los analistas y los clientes, para de esa forma garantizar que los requisitos que permanezcan luego de realizarle el análisis sean los correctos. Se propone que luego de obtener los requisitos se agrupen por las categorías antes expuesta y se prioricen según el siguiente criterio:

Crítico: Se refiere a las funcionalidades básicas del sistema.

Importante: Son los que brindan soporte a las funcionalidades básicas del sistema.

Útil: Son características que le ofrecen comodidad al sistema (tecnología multimedia). (2)

1.6.3 Especificación de Requisitos

Esta es la actividad en la que se registran los requisitos, se describen las funcionalidades del sistema que será desarrollado, se documentan diagramas, modelos del sistema y cualquier otro documento que sea importante y que se pueda utilizar en fases posteriores.

Existen diferentes técnicas para la Especificación de Requisitos dentro de las más usadas están:

Escenarios: la técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. La representación del escenario puede variar dependiendo del autor. Esta representación puede ser casi textual o ir encaminada hacia representaciones gráficas en forma de

Capítulo 1: Fundamentación Teórica

diagramas de flujo. El análisis de los escenarios, hechos de una forma u otra, pueden ofrecer información importante sobre las necesidades funcionales de sistema.

Glosario y ontologías: La diversidad de personas que forman parte de un proyecto de software hace que sea necesario establecer un marco de terminología común. Por esta razón son muchas las propuestas que abogan por desarrollar un glosario de términos en el que se recojan y definan los conceptos más relevantes y críticos para el sistema. En esta línea se encuentra también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos.

Lenguaje natural: Es flexible y sencillo aunque puede contener problemas de interpretación, por ejemplo pueden existir problemas de claridad a la hora de distinguir los requisitos o definir las metas del sistema. A pesar de que dicha técnica presenta estas problemáticas su uso se hace necesario con el fin de que los usuarios puedan comprender, sin necesidad de poseer conocimientos técnicos de desarrollo y comprobar que sus necesidades están expresadas en el documento de requisitos. (2)

1.6.4 Validación de Requisitos

La validación tiene como objetivo examinar las especificaciones para verificar que los requisitos del sistema han sido establecidos sin ambigüedad, inconsistencias, omisiones, que los errores detectados han sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso del proyecto y de producto.

Existen diferentes técnicas para la validación de requisitos estas son:

Auditorías: la revisión de la documentación con esta técnica consiste en el chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir, sólo una muestra es revisada.

Prototipos: Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

Capítulo 1: Fundamentación Teórica

Matrices de Trazabilidad: Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo los objetivos que cubre cada requisito para de esta forma detectar inconsistencias y objetivos no cubiertos.

Revisión: esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. (2)

1.6.5 Gestión de Requisitos

Con esta actividad se pretende llevar un control sobre los cambios que pueden sufrir los requisitos a lo largo del proceso. Para gestionar los requisitos se lleva a cabo un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento. (13)

Actividades para desarrollar la GR

Priorizar requisitos: Para determinar aquellos que deben cumplir en la primera versión del producto y aquellos que pueden llevarse a cabo en sucesivas versiones.

Gestión de cambios: Es vital gestionar estos cambios de forma efectiva y eficiente. Es adecuado que el documento de requisitos que se esté elaborando previo a entrar a la línea base esté sometido a un procedimiento de control de cambios para poder distinguir las versiones iniciales de las versiones aprobadas.

Realizar la trazabilidad: Es necesario tener buenas técnicas para separar y especificar correctamente los requisitos, controlar su evolución y soportar los cambios. La trazabilidad es el mecanismo que permite lograr este resultado. Esta proporciona elementos que ayudan a la comunicación entre los equipos de trabajo que es un factor que presenta gran problema en dicho proceso. (2)

Capítulo 1: Fundamentación Teórica

1.6 Metodologías de desarrollo de Software

Las metodologías describen un proceso disciplinado y detallado para el desarrollo de un software con el fin de hacerlo más eficiente, cumpliendo con las expectativas del cliente.

1.6.1 Rational Unified Process (RUP)

La metodología RUP es una metodología para la ingeniería de software que va más allá del simple análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software, se caracteriza por ser: Guiado por casos de uso, los mismos son el instrumento para validar la arquitectura del software y extraer los casos de prueba; centrado en la arquitectura, es donde los modelos son proyecciones del análisis y el diseño que constituyen la arquitectura del producto a desarrollar; iterativo e incremental, durante todo el proceso de desarrollo se producen versiones incrementales del producto en desarrollo. RUP se caracteriza por tener una amplia documentación. (2)

1.6.2 Extreme Programming (XP)

Esta metodología está basada en pruebas unitarias, estas son las pruebas realizadas a los principales procesos, consisten en comprobaciones (manuales o automatizadas) que se realizan para verificar que el código correspondiente a un módulo concreto de un sistema software funciona de acuerdo con los requisitos del sistema, es una metodología ágil, para entrega de productos en corto plazo, se caracteriza por la retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. Tiene como inconveniente que es una metodología que todavía une método con notación. (2)

1.6.3 Relationship Management Methodology (RMM)

Fue desarrollada en el año 1995. Se realiza la modelación de las aplicaciones a través de RMDM (Relationship Management Data Model), basado en el modelo Entidad – Relación y posee una herramienta CASE denominada: Relationship Management Case Tool – RMCASE.

RMM contiene el diseño y la construcción de aplicaciones hipermedia en un proceso de siete pasos. RMM especifica la navegación con primitivas de acceso, como enlaces (links), agrupamiento (menus), índices (index) y recorridos guiados (guided tours). La técnica propuesta para el diseño de la interfaz de usuario

Capítulo 1: Fundamentación Teórica

es la elaboración de maquetas y prototipos. Tiene como inconveniente que al igual que Extreme Programming (XP), un método con notación. (1)

1.6.4 Marco de trabajo

Todo Proceso de Desarrollo de Software define un Marco de trabajo común el cual está compuesto por dos grupos, un primer grupo definido por las actividades genéricas o comunes y un segundo grupo definidos por las actividades sombrilla, ambas formadas por un conjunto de acciones las cuales están compuestas por un conjunto de tareas que expresan de que forma va a ocurrir ese Marco de trabajo.

Las actividades genéricas o comunes del proceso son las siguientes: Comunicación, Planificación, Modelado, Construcción y Despliegue.

- **Comunicación:** esta actividad implica una intensa comunicación y colaboración con el cliente; además abarca la investigación de requisitos realizada.
- **Planeación:** esta actividad establece un plan para el trabajo, describe las tareas técnicas que deben realizarse, los riesgos probables y un programa de trabajo.
- **Modelado:** esta actividad abarca la creación de modelos que permiten al desarrollador y al cliente comprender los requisitos del software y el diseño que logrará satisfacerlo.
- **Construcción:** esta actividad combina la generación del código y la realización de pruebas necesarias para descubrir errores en el código.
- **Despliegue:** el software se entrega al cliente, quién evalúa el producto recibido y proporciona información basada en su evaluación.(13)

Las actividades sombrilla son las que realizan un mejor control, seguimiento y ejecución del proceso, brindándole una mejor calidad y completitud en el proceso de desarrollo del software. Dentro de estas se encuentran:

- **Seguimiento y control del proyecto de software:** Permite que el equipo de software evalúe el progreso comparándolo con el plan de proyecto y así tomar las acciones necesarias para mantener el programa.

Capítulo 1: Fundamentación Teórica

- **Gestión del riesgo:** Evalúa los riesgos que pudieran afectar los resultados del proyecto o la calidad del proyecto.
- **Aseguramiento de la calidad:** Define y conduce las actividades requeridas para asegurar la calidad del software.
- **Revisiones técnicas formales:** Está encaminado a descubrir y eliminar los errores antes que estos se propaguen Hacia la siguiente acción o actividad. (13)

Con la realización de este estudio se llegó a la conclusión que todas las metodologías están basadas en características particulares que para el tipo de aplicación que se está realizando, el entorno de desarrollo con el cliente y las características del equipo de desarrollo, no se ajustan a estas metodologías. RUP es utilizado para proyectos grandes, tiene 4 fases y 9 flujos de trabajo lo cual la convierte en un proceso largo, por lo que no conviene ser uso de ella cuando se dispone de poco tiempo para la entrega del producto y las metodologías XP y RMM todavía unen método con notación, lo que trae como desventaja que se obliga a utilizar el lenguaje de modelado que fue definida. Por lo que se utilizará un Marco de trabajo que describe las actividades para la realización del Laboratorio Virtual de Sistemas Operativos.

Se utilizará para el desarrollo del trabajo un enfoque ágil, porque el entorno y el equipo de trabajo se ajustan a las características de este enfoque, que aunque no se cuenta con una abundante documentación, Todo el equipo de trabajo utiliza un enfoque común, existen habilidades para la toma de decisiones y organización propia.

La siguiente tabla refleja las actividades genéricas del Marco de trabajo con las acciones que se van a realizar en cada una de estas actividades.

Actividades Genéricas	Acciones
Comunicación	Entrevistas con el cliente. Extracción de requisitos. Especificación de requisitos.

Capítulo 1: Fundamentación Teórica

Planificación	Plan de trabajo que contiene las tareas que fueron realizadas.
Modelado	Diagrama de Casos de Uso. Diagramas de Actividades. Descripciones Textuales de Casos de Uso. Diagramas de Clase. Diagramas de Navegación. Diagramas de Presentación. Diagramas de Secuencia.
Construcción	
Despliegue	

Tabla 1: Acciones por las que están compuestas las actividades genéricas.

1.7 Herramientas CASE

Las herramientas CASE favorecen el apoyo al desarrollo de software, proporcionando la modelación de un conjunto de diagramas que describen todo el proceso en la Ingeniería de Software durante todo el ciclo de vida del desarrollo del sistema.

1.7.1 Rational Rose Enterprise Edition

Rational Rose es la herramienta CASE desarrollada por los creadores de UML, que cubre todo el ciclo de vida de un proyecto, desde la fase de inicio, formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Dentro de sus características principales se puede encontrar un avanzado modelado de UML para trabajar en diseños de bases de datos, con capacidad de representar la integración de los datos y los requisitos de aplicación a través de diseños lógicos y físicos,

Capítulo 1: Fundamentación Teórica

posee además una fuerte capacidad para integrarse con cualquier sistema de control de versiones. Es importante resaltar que a pesar de esto Rational Rose presenta una necesidad de alta capacidad de procesamiento y se necesita tener habilidad y conocimiento de esta herramienta para trabajar con ella. (9)

1.7.2 Enterprise Architect

Es una herramienta multi-usuario que combina el poder de la especificación UML 2.1 y BPMN 1.1, con alto rendimiento y una interfaz intuitiva. Está basada en Windows y fue diseñada para ayudar a construir un software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. Otro elemento importante es el bajo costo de sus licencias. Entre las ventajas que ofrece Architect se encuentran: Alta capacidad, soporta control de versiones, velocidad y buen rendimiento entre otras.

1.7.3 Visual Paradigm

Visual Paradigm para UML es una de las herramientas UML CASE más completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. VP también permite realizar ingeniería inversa, es una herramienta que tiene licencia.

Concluido el estudio se decidió que la herramienta a utilizar para el diseño es Visual Paradigm ya que es de las más potentes que existen, es una herramienta multiplataforma, que tiene versiones libre, en la que se pueden construir diferentes tipos de diagramas, lo que le confiere un alto grado de usabilidad. La universidad cuenta además con la licencia para su utilización legal en la universidad. (3)

1.8 Lenguaje de Modelado

1.8.1 Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia (OMMMA – L).

Fue desarrollado tomando como base el lenguaje UML. Consta de cuatro vistas fundamentales en su modelación: vista lógica, vista de presentación espacial, vista de comportamiento temporal predefinido y vista de control interactivo. Modifica los diagramas originales de UML de: clases, secuencia y estado. Añade como parte de la vista de presentación espacial un nuevo diagrama: el diagrama de presentación,

Capítulo 1: Fundamentación Teórica

para la representación espacial de los elementos visuales del futuro software multimedia. Basa su descripción en el patrón de arquitectura MVCMM.

1.8.2 El Lenguaje de Modelado Unificado (UML)

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Ayuda a la captura de decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Incluye conceptos semánticos, notación, y principios generales. (1)

1.8.3 Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)

ApEM-L es un Lenguaje para la Modelación de Aplicaciones Educativas, se presenta como una extensión de UML, tomando como bases teóricas principales OMMMA – L (2001) y OCL – 2.0 (2003), lo que produce las siguientes ventajas:

- Puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML.
- No modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base. (1)

La concepción del Lenguaje para la Modelación de Aplicaciones Educativas (ApEM – L) tuvo los siguientes objetivos:

- Desarrollar una extensión del lenguaje de modelado UML, tomándolo como base e incorporando a este, a través de sus mecanismos de extensión, los elementos fundamentales del proceso productivo UCI. De esta forma se produjo un lenguaje de propósito particular para la modelación de aplicaciones educativas.

Capítulo 1: Fundamentación Teórica

- Incorporar los elementos más significativos de extensiones anteriores como OMMMA – L (2001) y a su vez respetar lo establecido por el estándar OCL (2003), para de esta forma lograr una extensión consistente y escalable en el tiempo.
- No circunscribir ApEM-L a un proceso de desarrollo en específico, sino expresarlo de manera tal que pueda ser utilizado con cualquiera de los existentes, aunque se sugiere la utilización de procesos de desarrollo iterativo e incremental, que permitan la modelación de sistemas orientados a objetos.

ApEM-L consta de tres áreas fundamentales: de estructura lógica, de comportamiento dinámico y de gestión del modelo, y a su vez de cuatro vistas distribuidas en dichas áreas: estática, de arquitectura, de comportamiento y de presentación; para la modelación de los productos, a través de un conjunto de diagramas distribuidos por cada una de estas vistas. (1)

Después de un estudio de los lenguajes de modelado se detectó que UML y OMMMA-L no logran denotar todos los elementos, tanto ingenieriles como pedagógicos de las aplicaciones educativas cubanas, no soporta todos los aspectos de las aplicaciones multimedia de una forma adecuada e intuitiva. Especialmente, las características del lenguaje para modelar los aspectos de la interfaz de usuario, no se aplican explícitamente en los entornos multimedia. (1)

Por lo anteriormente planteado se decide escoger ApEM-L, la cual satisface las deficiencias detectadas en los dos lenguajes expuestos anteriormente, además que incorporan estereotipos descriptivos y restrictivos.

1.9 Patrones

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Existen patrones de casos de uso que ayudan a la confección del diagrama de casos de uso, patrones arquitectónicos cuyo objetivo es definir la arquitectura del sistema y los patrones de diseño los cuales contribuyen a las buenas prácticas de la programación.

1.9.1 Patrones de Casos de Uso

Capítulo 1: Fundamentación Teórica

Inclusión Concreta y Extensión Concreta: Un caso de uso extiende a otro cuando sin alterar a este, se incorpora su funcionalidad como parte integral del primero. Se denota con una relación que apunta del caso extendido al caso base y la conexión se hace o bien al principio del flujo de eventos principal del caso base o en alguno de los puntos de extensión que este haya definido.

La relación de extensión guarda relación con todos los restantes tipos de reutilización que están definidas para los casos de uso; en particular la relación es muy íntima con la relación de inclusión <<include>>, sin embargo la diferencia primordial entre <<extend>> e <<include>> es la modificación del caso base. Extensión no implica cambio, en tanto que la inclusión añade funcionalidad al caso base.

Múltiples actores (Roles diferentes): Captura la concordancia entre actores manteniendo roles separados. Consiste de un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el caso de uso.

Concordancia de Adición: En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema. (10)

CRUD (Creating, Reading, Updating, Deleting): Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Completo: Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

Parcial: Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras. (14)

1.9.2 Patrón Arquitectónico

Capítulo 1: Fundamentación Teórica

El patrón Modelo – Vista – Controlador – Entidad (**MVC-E**) como propuesta arquitectónica del software educativo.

Son necesarias las visiones esenciales y de implementación del software para acomodar las restricciones lógicas impuestas por los requisitos del procesamiento y las restricciones físicas impuestas por otros elementos del sistema.

Este enfoque unido a las características distintivas del software educativo producido en Cuba ha traído como consecuencia el análisis de una variante de solución al MVCMM para las aplicaciones educativas cubanas, donde se descarguen las responsabilidades de la clase modelo concernientes al procesamiento y almacenamiento de la información persistente de las aplicaciones, incorporando una nueva clase al modelo denominada Modelo Entidad, con dos tipos fundamentales, la **clase Modelo-Entidad-Media** y **Modelo-Entidad-Persistente**.

La primera de estas con la responsabilidad de agrupar las clases que identifican las medias y su árbol de jerarquía en la aplicación y la segunda tienen como responsabilidad la gestión de la información persistente, que antes sobrecargaba a la clase Modelo del patrón MVC original. Esta variación a su vez sustenta las características actuales de los sistemas multimedia como es la comunicación con bases de datos. (1)

1.9.3 Patrones de Diseño

Dentro de los patrones de diseño se encuentran los patrones GOF y GRASP.

Patrones GOF:

Singleton: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

Fábrica Abstracta: Permite a un sistema determinar la subclase a partir de la cual se va a instanciar un objeto en tiempo de ejecución. A menudo, esta subclase es desconocida durante el desarrollo.

Visitor: Representa una operación sobre los elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera.

Capítulo 1: Fundamentación Teórica

Observer: Es una forma de notificar cambios a diferentes clases dependientes. (11)

Patrones GRASP:

Bajo Acoplamiento: El acoplamiento mide qué tan fuerte está una clase conectada con otras (es decir, cuántas clases conoce y necesita). Una clase con bajo acoplamiento no depende de "muchas otras" clases. Una clase con alto acoplamiento recurre a muchas otras clases. Este tipo de clase no es conveniente, pues: cambios en las clases relacionadas ocasionan cambios en la clase local; son más difíciles de entender; son más difíciles de reutilizar.

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

Alta cohesión: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Controlador: Asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. (14)

1.13 Conclusiones del Capítulo

Con el estudio del capítulo se analizó la importancia hoy día de los Laboratorios Virtuales ya que contribuyen al aprendizaje colaborativo y permite la comunicación entre usuarios, haciendo posible la transmisión de experiencias y conocimientos. Se valoró las herramientas existentes para el diseño del Mapa conceptual donde el cliente estableció que fuera CompendiumLD. Con el estudio de la Ingeniería de Requisitos, se logró determinar cuáles serían las técnicas utilizadas para la Extracción, Especificación, Validación y Gestión de los Requisitos. También con el estudio de las metodologías se llegó a la conclusión que ninguna cumplía en su totalidad con las actividades que se debían realizar para el proceso pedagógico, por lo que se utiliza un Marco de trabajo. Una vez estudiado las herramientas CASE y los lenguajes de modelado se decide utilizar Visual Paradigm y ApEM- L las cuales permiten la confección de

Capítulo 1: Fundamentación Teórica

los diferentes diagramas, se estudiaron además los patrones y las técnicas. Sentándose las bases para comenzar la realización de la solución que a continuación se propone.

Capítulo 2: Solución Propuesta

El Mapa Conceptual mostrado anteriormente representa todo el contenido referente al tema de Exclusión Mutua y Sincronización de procesos, donde los procesos tienen condiciones de competencia que definen una sección crítica las cuales utilizan Exclusión Mutua que se solucionan con diferentes condiciones (Exclusión Mutua, Suposiciones, Progreso y Límite de tiempo). La Exclusión Mutua puede ser de dos tipos de Espera ocupada o de Bloqueo, la primera incluye Desactivación de interrupciones, Alternancia estricta y Solución de Peterson, la segunda utiliza Semáforos que definen primitivas P() y V(). En Exclusión Mutua con bloqueo se modela estructuras Fork/Join y Parbegin/Parend.

2.2 Requisitos funcionales y no funcionales

Después del estudio realizado en el capítulo anterior de las técnicas existentes en la Extracción de Requisitos, se seleccionó la técnica de la entrevista abierta porque no se formula las preguntas en documentos, y la tormenta de ideas para el debate logrando de este modo obtener criterios con respecto a lo que se quiere realizar, dentro de las técnicas de Especificación de Requisitos se escogió el Lenguaje Natural para que fuera entendible por cualquier persona. Gracias a estas técnicas se logró identificar y describir los siguientes requisitos funcionales y no funcionales:

Requisitos Funcionales

RF 1: El sistema debe permitir visualizar el contenido que compone el proceso Exclusión Mutua y Sincronización.

RF 2: El sistema debe permitir modificar el contenido.

RF 3: El sistema debe permitir deshabilitar el contenido.

RF 4: El sistema debe permitir habilitar el contenido.

RF 5: El sistema debe permitir estudiar contenidos.

RF 5.1: El sistema debe permitir reanudar lección.

RF 5.2: El sistema debe permitir seleccionar el contenido a estudiar.

RF5.3: El sistema debe permitir guardar las trazas de navegación.

RF 5.4: El sistema debe permitir culminar su navegación.

Capítulo 2: Solución Propuesta

RF 6: El sistema debe permitir consultar su evaluación.

RF 7: El sistema debe permitir revalorizar evaluación.

RF 8: El sistema debe permitir rechazar ejercicio.

RF 9: El sistema debe permitir solicitar ayuda.

RF 10: El sistema debe permitir Abandonar tema.

Requisitos no funcionales

Apariencia o interfaz externa:

RNF1: El diseño de la interfaz debe ser sencillo y fácil de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones.

RNF2: La combinación de colores debe ser agradable a la vista del usuario.

Usabilidad:

RNF3: El sistema puede ser usado por cualquier persona que posea conocimientos básicos sobre el funcionamiento interno de un Sistema Operativo.

Rendimiento:

RNF4: La respuesta a solicitudes más complejas de los usuarios del sistema no debe exceder 9 segundos.

Disponibilidad

RNF5: La aplicación deberá estar disponible las 24 horas del día.

Portabilidad:

RNF6: Debe poder ejecutarse en Sistemas Operativos superior a Windows XP 98 y Ubuntu 10.10.

Seguridad:

Capítulo 2: Solución Propuesta

RNF7: Garantizar que la información sólo sea modificada por las personas que tienen permisos para realizar esta actividad.

Legales.

RNF8: Cada una de las medias (imágenes, videos, sonidos) que se utilicen en el producto deben tener el permiso legal de sus autores y su aprobación para hacer uso de ellas.

2.3 Diagrama de Casos de Uso

En el siguiente diagrama de caso de uso se puede identificar cuáles son los usuarios que existen y cuáles son las acciones que podrán realizar según el permiso que se halla establecido.

Los patrones de casos de uso que se ponen de manifiesto son:

- Exclusión Concreta.
- Inclusión Concreta.
- Concordancia de Adición.

Capítulo 2: Solución Propuesta

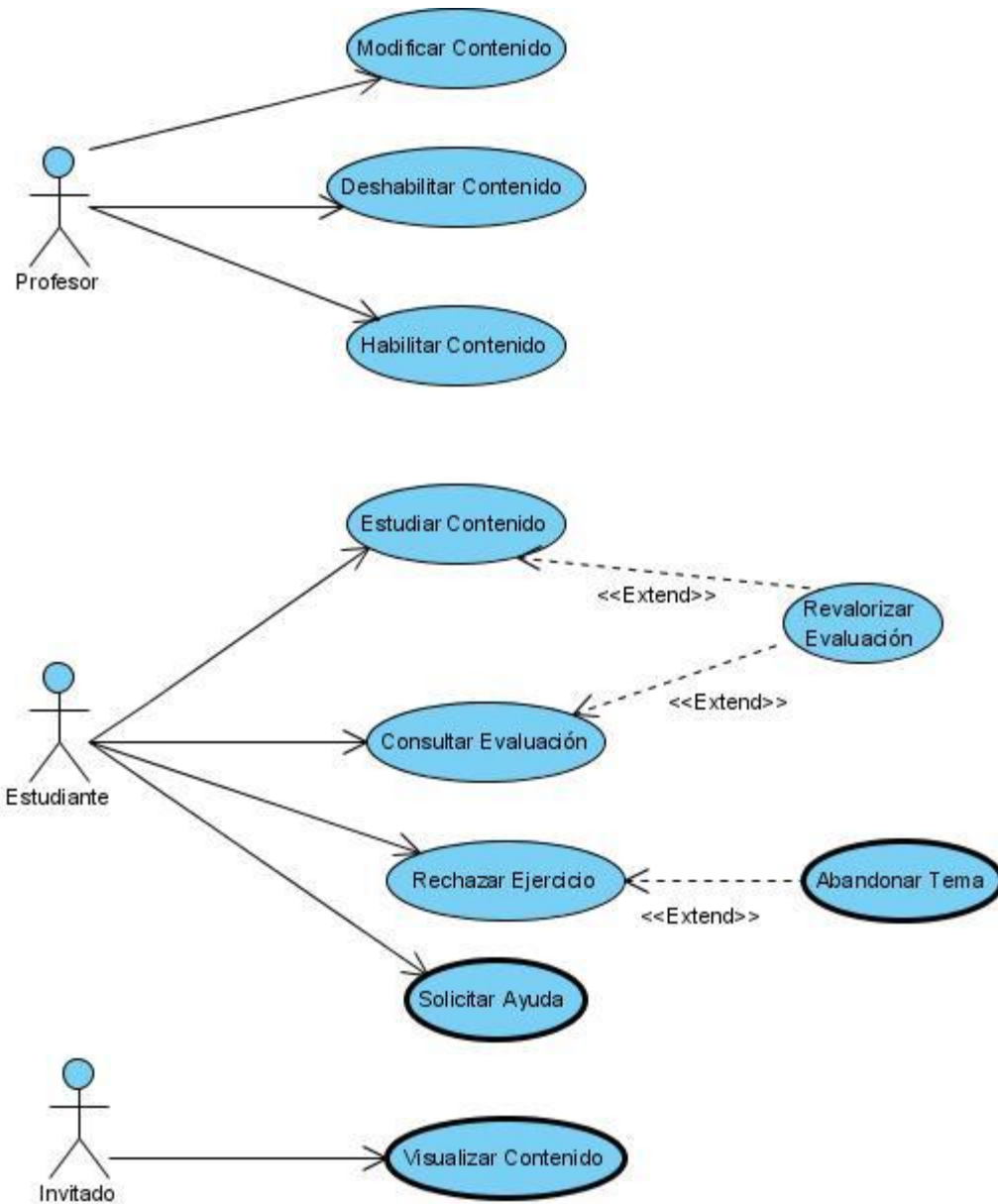


Figura 2: Diagrama de Caso de Uso

2.4 Descripción Textual

A continuación se muestra la Descripción Textual del caso de uso Habilitar Contenido con todo su flujo.

Capítulo 2: Solución Propuesta

Descripción textual del caso de uso “Habilitar Contenidos”	
Actores del caso de uso	Profesor (Inicia)
Propósito	El caso de uso tiene como propósito permitir al profesor habilitar contenidos.
Resumen	El caso de uso se inicia cuando el profesor selecciona la opción “Habilitar Contenidos”, una vez seleccionada la opción deseada el profesor decide que contenido desea habilitar, luego selecciona la opción “Deshabilitar” y culmina el caso de uso cuando se presenta el Menú Principal.
Casos de uso asociados	
Referencias	RF4
Precondiciones	El actor debe estar autenticado en el sistema.
Poscondiciones	Los contenidos quedan habilitados para los estudiantes.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema

Capítulo 2: Solución Propuesta

1. El Profesor selecciona la opción “Habilitar Contenido”.	1.1 El sistema verifica si hay contenidos deshabilitados. 1.2 El sistema muestra el listado de contenidos disponibles a habilitar
2. El Profesor elige uno.	
3. El profesor selecciona la opción “Aceptar”.	3.1 El sistema chequea las reglas de precedencia. 3.2 Si cumple las condiciones entonces habilita el contenido. 3.3 Muestra nuevamente el listado de los contenidos a habilitar.
Cursos Alternos de los Eventos	
Acción	Curso Alterno
1.1	Si no existen contenidos para deshabilitar el sistema le muestra un mensaje (no existen contenidos disponibles a habilitar).
3.	El Profesor selecciona la opción Cancelar, muestra el menú principal y culmina el caso de uso.
3.2	Si no cumple con las reglas de precedencia muestra un mensaje al profesor que no cumple con las reglas de precedencia y muestra el menú Principal y culmina el caso de uso.
Prioridad	Crítico
Mejoras	

Capítulo 2: Solución Propuesta

Medias a utilizar	Tipo de Media	Descripción	Estado
	Imagen	Fondo del menú principal y se usará imagen además para el fondo del listado de los contenidos en general y para cada contenido por separado.	En construcción
	Video o Animación	No tiene	
	Sonido	Sonido de pequeña duración para cuando se pase por encima de las opciones del menú principal y de la clase con los contenidos a habilitar.	En localización
	Texto	Texto de la clase menú Principal, la de los contenidos generales y de cada uno en específico, además de un texto asociado a cada botón.	En construcción
Elementos pedagógicos			

Tabla 2: Descripción textual del caso de uso Habilitar Contenido.

Capítulo 2: Solución Propuesta

2.5 Diagrama de Actividades

Los diagramas de actividades representan los flujos de eventos. Estos diagramas permiten visualizar mejor los flujos y ayudan al entendimiento del equipo de desarrollo.

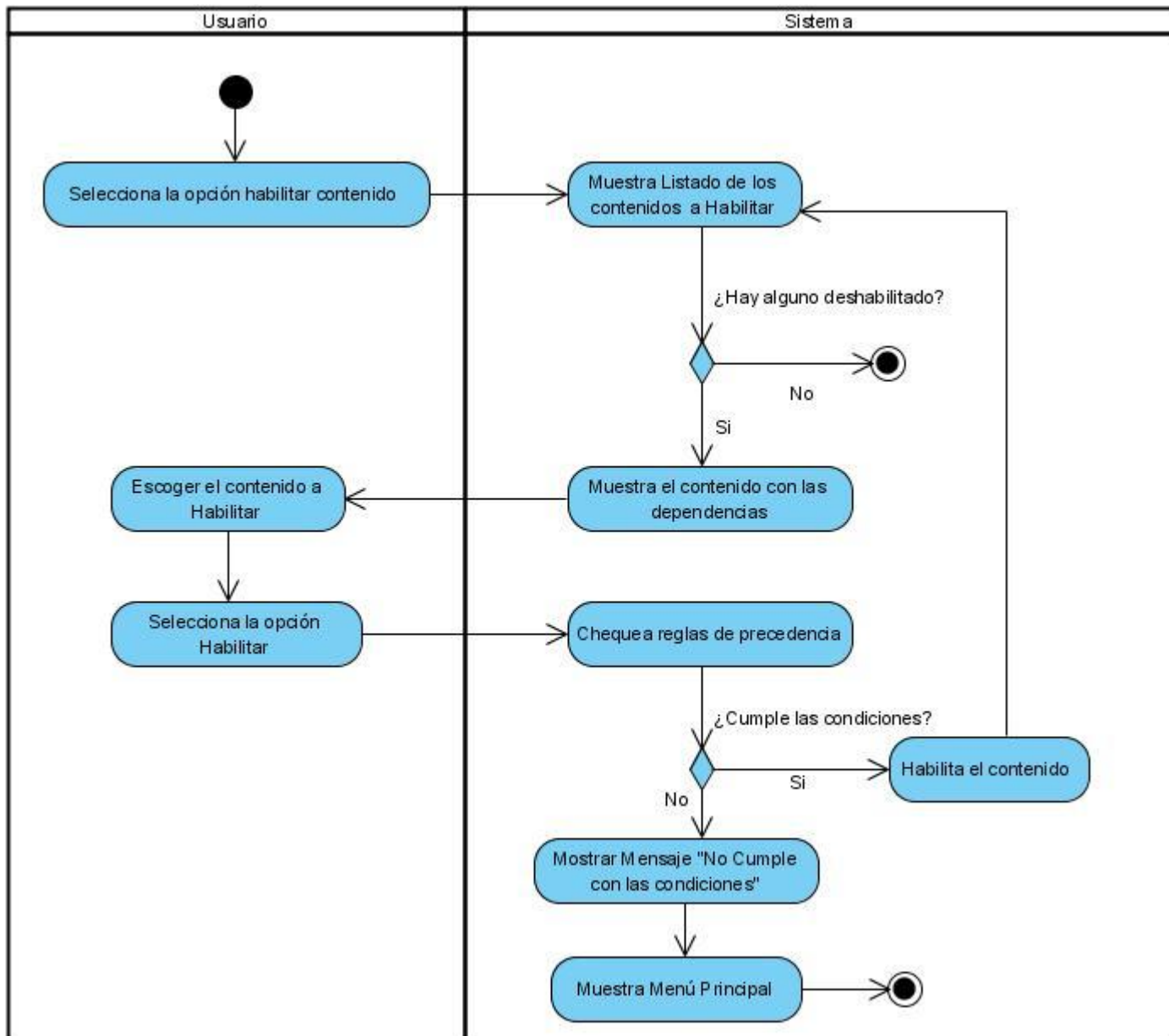


Figura 3: Diagrama de Actividades del caso de uso Habilitar Contenido.

Capítulo 2: Solución Propuesta

Diagrama de Clases

El diagrama de clases está compuesto por las siguientes Clases:

- **Clases Modelo Entidad Medias:** Correspondientes a la representación del árbol jerárquico de medias.
- **Clases Vista:** Recibirán las peticiones del usuario al sistema y mostraran los mensajes de salida o respuestas.
- **Clases Controladoras:** Gestionarán las peticiones y la muestra de las respuestas.
- **Clases Modelo:** Es la que contendrán la lógica de negocio para el procesamiento de la información.
- **Clases Modelo Entidad:** Esta clase es para el procesamiento de la información persistente.
- **Clase HLL:** Es la clase correspondientes al lenguaje de alto nivel con el que se programe.

Los patrones de diseño que se ponen de manifiesto son:

Experto: Es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo, esta clase concedora de información es la clase Modelo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Alta cohesión: La información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.

Singleton: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. Esto se utiliza con las clases persistentes del sistema las cuales contienen toda la información persistente de la aplicación.

Capítulo 2: Solución Propuesta

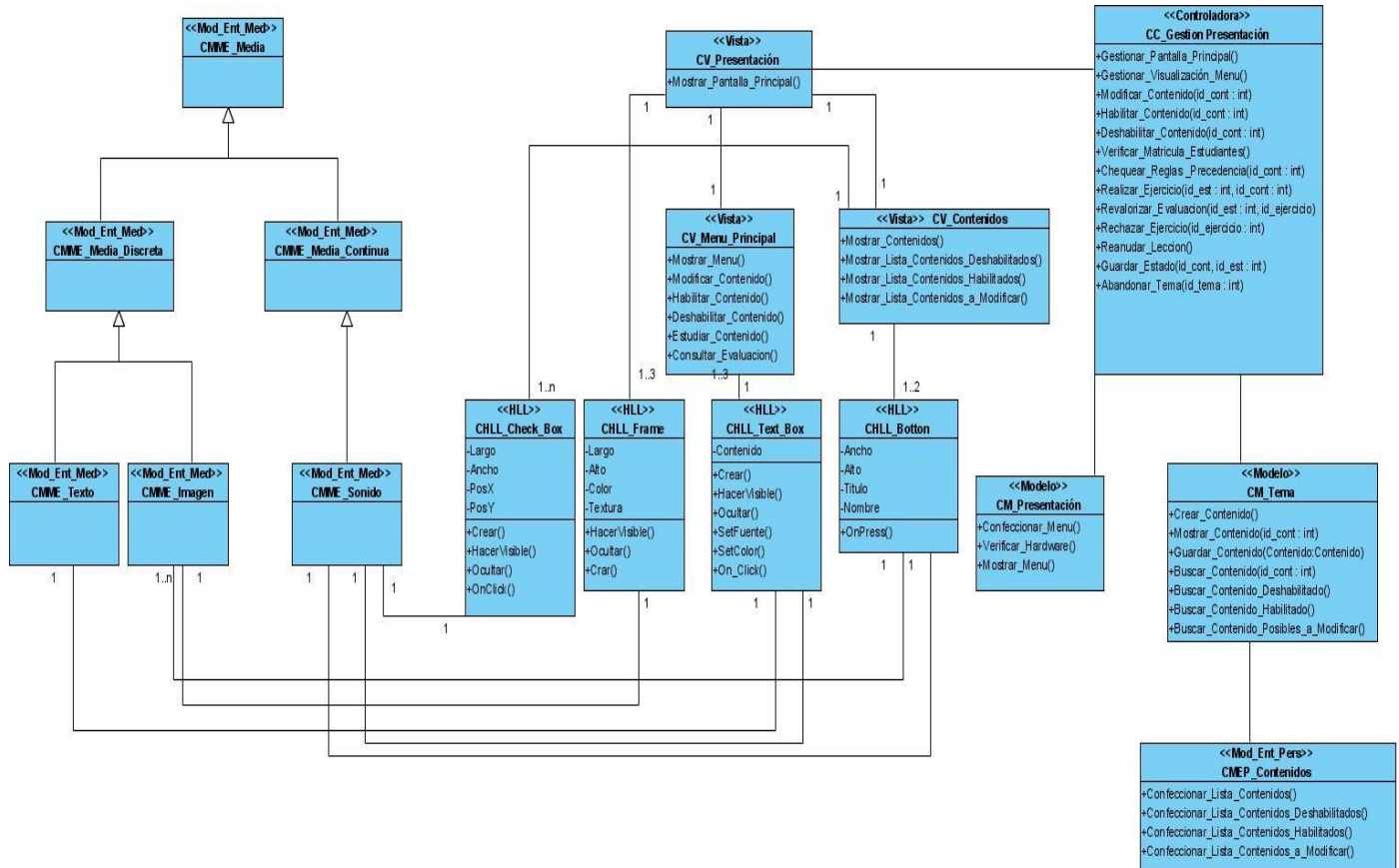


Figura 4: Diagrama de Clases del caso de uso Habilitar Contenido.

2.5 Diagrama de Navegación

Los diagramas de navegación están compuestos por las siguientes clases:

- **Clase menú (<<Menu>>):** Es el elemento de composición de una clase vista, desde donde se puede llegar a otras diferentes clases vistas con las cuales se conecta este menú, pues contiene una lista de las opciones de movimiento siguiente, pero no los valores de esas opciones, los cuales serán ofrecidos por el resto de los tipos de clases. Dicho menú no tiene que necesariamente llevar a una clase vista directamente, sino que puede ser a través de otro tipo de clase de navegación (guía, consulta o índice).

Capítulo 2: Solución Propuesta

- **Clase consulta (<<Consulta>>):** Es aquel tipo de clase que permite el enlace con una clase vista a y través de un valor directo (variable de consulta) asignado a la búsqueda para la visualización del elemento a mostrar, que se sitúa como identificador en la relación entre las clases.
- **Clase índice (<<Indice>>):** Es aquella clase que denota el valor de direccionamiento hacia una clase vista a partir de una opción determinada.
- **Clase botón (<<Boton>>):** Es aquella clase que denota la existencia de un elemento interactivo del tipo botón para producir un camino en la navegación hacia una clase vista, y utilizando como intermediarias a clases de tipo consulta o índice.

Capítulo 2: Solución Propuesta

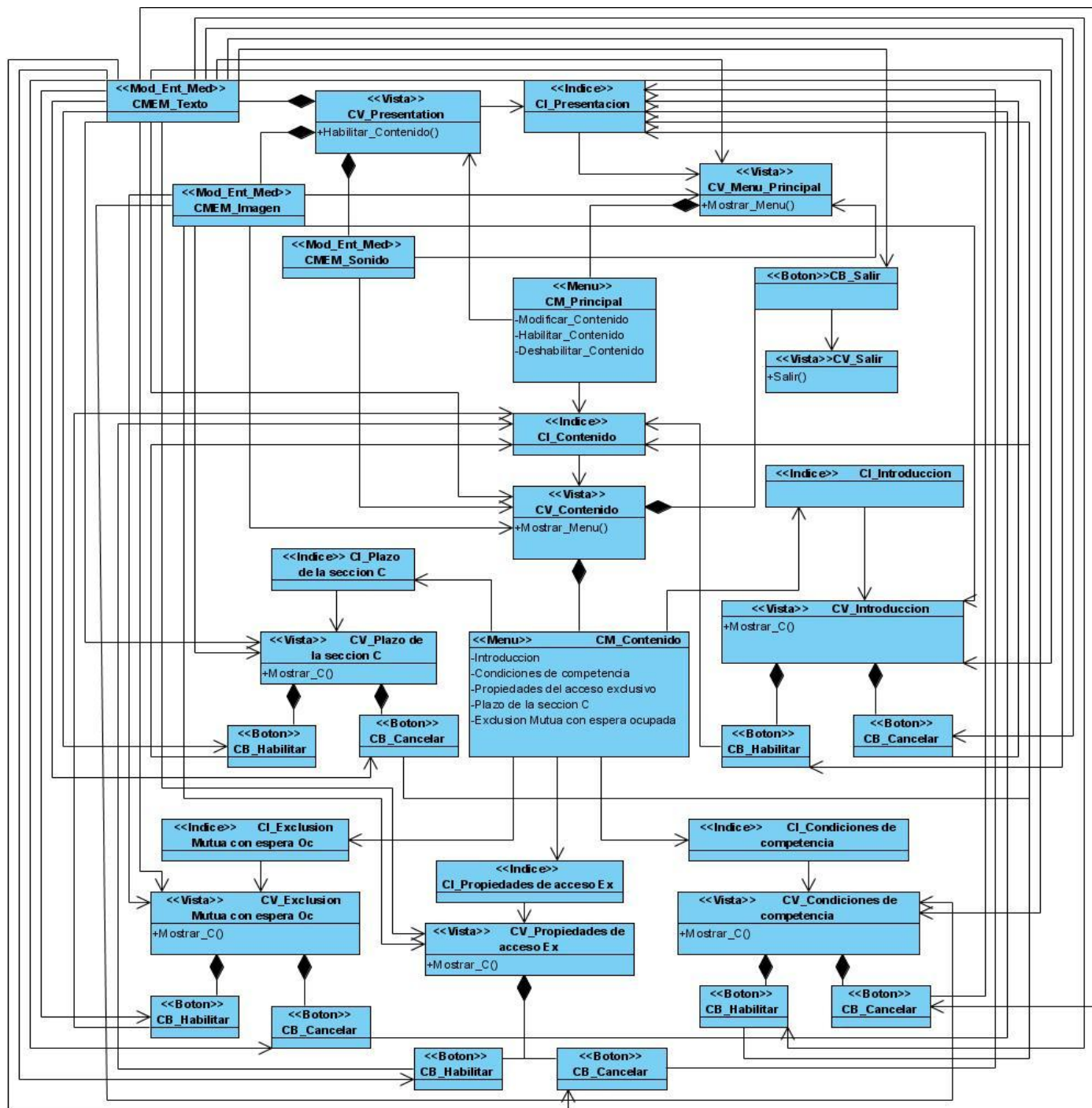


Figura 5: Diagrama de Navegación del caso de uso Habilitar Contenido.

Capítulo 2: Solución Propuesta

2.6 Diagrama de Presentación

Los diagramas de presentación están compuestos por dos clases las estáticas y la interacción.

- **Clase estáticos:** Agruparán los componentes que solo tiene como función visualizar información, pero que no permiten interacción con el usuario.
- **Clase interacción:** Las cuales serán los elementos de la vista que permiten interacción del usuario con el sistema informático modelado.

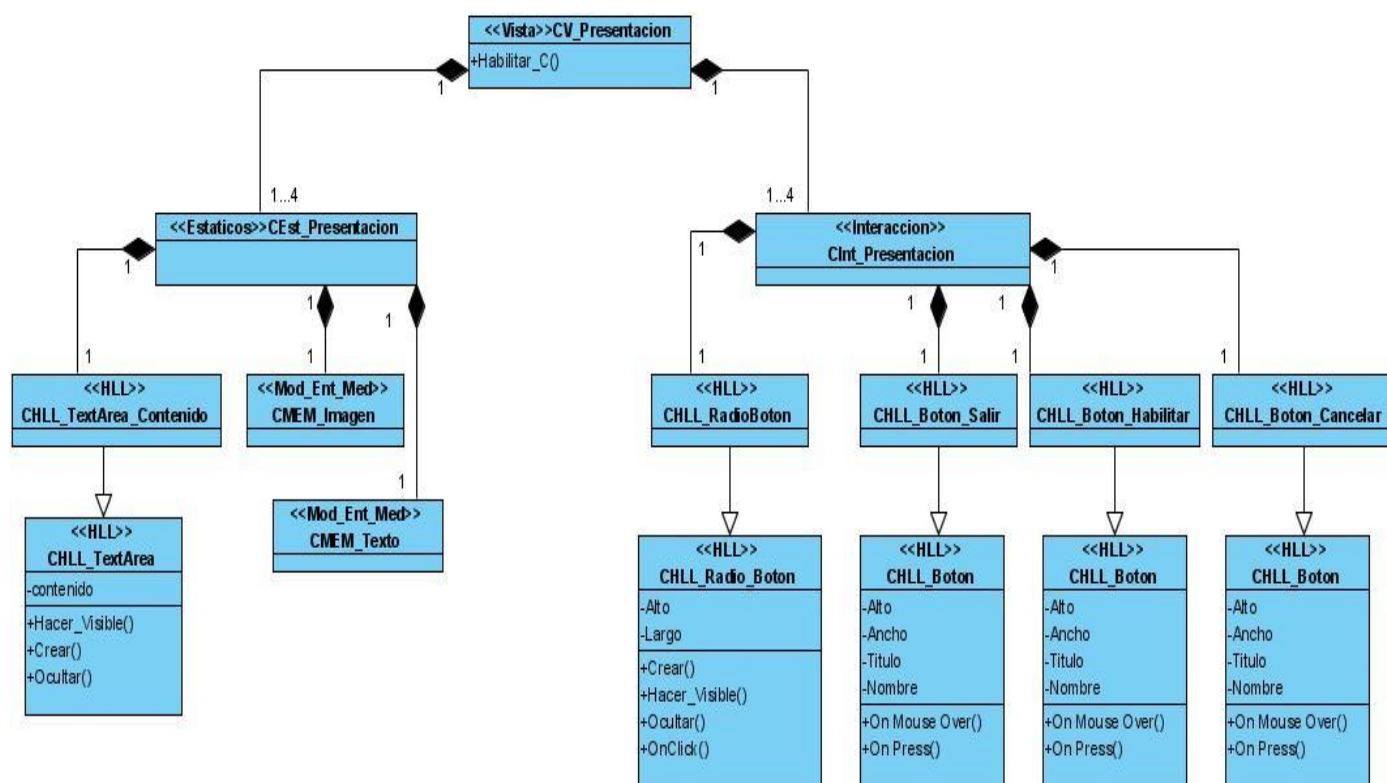


Figura 6: Diagrama de Presentación del caso de uso Habilitar Contenido.

Capítulo 2: Solución Propuesta

2.7 Diagrama de Secuencia

Los diagramas de Secuencia representan la manera en la que los objetos de la aplicación intercambian mensajes para darle cumplimiento a sus responsabilidades.

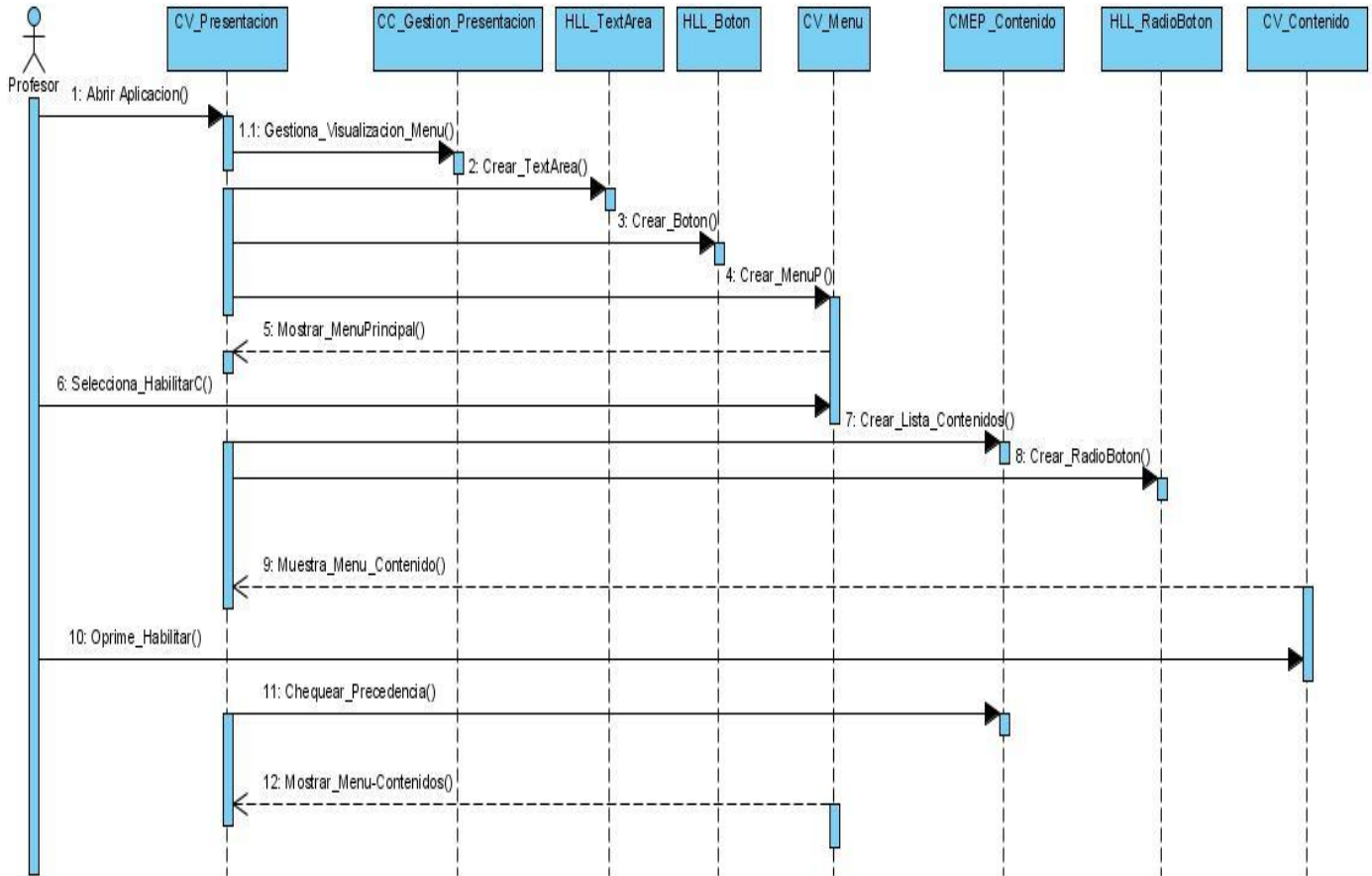


Figura 7: Diagrama de Secuencia del caso de uso Habilitar Contenido.

2.8 Conclusiones del capítulo.

Este capítulo permitió la creación del Mapa conceptual con la herramienta CompendiumLD, lo que facilitó el entendimiento del módulo Exclusión Mutua y Sincronización. Se obtuvieron los requisitos funcionales y no funcionales del sistema mediante las técnicas de Extracción de Requisitos seleccionadas, del mismo modo con la herramienta Visual Paradigm se realizó el diagrama de casos de uso del sistema poniendo en práctica los patrones de casos de uso que fueron necesarios para la confección del mismo, además de la

Capítulo 2: Solución Propuesta

descripción textual de los casos de uso y todos los diferentes tipos de diagramas, quedando de esta forma realizada la Ingeniería de Requisitos y el diseño a del módulo Exclusión Mutua y Sincronización.

Capítulo 3: Validación

Capítulo 3 Validación

3.1 Introducción.

En este capítulo se realizará la validación de la solución propuesta, específicamente se validará la Especificación de Requisitos y el diagrama de Caso de Uso del sistema para esto se utilizará Métricas de calidad, además se validará el diseño con las Métricas de tamaño operacional de clase y Relaciones entre clases.

3.2 Validación de Requisitos mediante Métricas de la calidad para la especificación.

Las Métricas de calidad para la especificación es una técnica que da una medida de cuán buena es una especificación de requisitos, las mismas estableciendo magnitudes y valores que se podrán deducir de la Especificación de Requisitos.

Mediante las métricas se comprueba que la especificación de requisitos sea correcta, que no existan requisitos ambiguos. Para esto es necesario conocer el total de los requisitos **Rt** dado por:

$$Rt = Rf + Rnf$$

$$Rt = 14 + 8$$

$$Rt = 22$$

Donde:

Rt: Total de Requisitos.

Rf: Cantidad de requisitos funcionales.

Rnf: Cantidad de requisitos no funcionales.

Para determinar la **especificidad (ausencia de ambigüedad)** se calcula **Q1** para determinar la especificidad de los requisitos.

$$Q1 = Rui / Rt$$

$$Q1 = 21 / 22$$

$$Q1 = 0.95$$

Donde:

Rui: Número de requisitos para los que todos los revisores tuvieron interpretación idéntica.

Q1: Ausencia de ambigüedad.

Capítulo 3: Validación

Es importante destacar que mientras más cerca de 1 esté Q1, menos será la ambigüedad de la especificación. (9)

Para la realización de esta Métrica se contó con el apoyo de personas con conocimientos del tema como: Ing. Aray Pérez Degue.

Ing Yinett Hernández Hernández.

Ing. Carlos Yasmany Hidalgo García.

Ing. Susana Fadragas Rodríguez.

Ing. Yelena Hernández Estrada.

Arribó el siguiente resultado:

El valor de **Q1** es de 0.95, esto indica que los requisitos tienen un Alto nivel de especificidad.

3.3 Modelo de Métricas OO aplicadas al Diagrama de Casos de Uso del Sistema.

Para medir la calidad del diagrama de casos de uso del sistema se deben tomar en cuenta los siguientes aspectos:

Compleitud: Grado en el que se han detallado los casos de uso más relevantes.

Consistencia: Grado en que los casos de uso del sistema describen las interacciones entre el sistema y los usuarios.

Correctitud: Grado en que las interacciones entre los actores y el sistema soportan adecuadamente el modelo de negocio.

Complejidad: Grado de claridad en la presentación de los elementos que describen la claridad y el contexto del sistema.

Cada uno de estos atributos tendrán asociada una o varias métricas que establecen un valor cuantitativo del grado en que los factores presentan un nivel Alto, Medio o Bajo (9)

Atributo	Factor	Métrica Asociada	Valor
Compleitud	Factor 1. ¿Han sido definidos todos los roles relevantes del usuario encargado de	Métrica 1. Número de roles relevantes omitidos. Umbral<10	Total de roles: 3 Número de roles relevantes omitidos: 0 Representa: 0%.

Capítulo 3: Validación

	generar/modificar o consultar información?		
	Factor 2. ¿Se presenta una descripción resumida de todos los casos de uso?	Métrica 2. Número de casos de uso que no tiene descripción resumida. Umbral<10	Total de casos de uso:10 Número de casos de uso que no tiene descripción resumida: 0 Representa: 0%.
	Factor 3. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	Métrica 3. Numero de requisitos omitidos por casos de uso. Umbral<15	Total de requisitos funcionales: 14 Número de requisitos omitidos por casos de uso: 1 Representa: 7.14 %.
		Métrica 4. Número de casos de uso que tienen requisitos omitidos. Umbral<15	Total de casos de uso: 10 Número de casos de uso que tienen requisitos omitidos: 1 Representa: 10 %.
	Factor 4. ¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia en (crítico, importantes y útiles)?	Métrica 5. Número de casos que no han sido clasificados. Umbral<10	Total de casos de uso: 10 Número de casos que no han sido clasificados: 0 Representa: 0%.
			Para un incumplimiento: 2.1 %
Consistencia	Factor 5. ¿El nombre dado a los casos de uso	Métrica 6. Número de casos de uso que	Total de casos de uso:11 Número de casos de uso

Capítulo 3: Validación

	es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	tienen un nombre incorrecto. Umbral<10	que tienen un nombre incorrecto: 0 Representa: 0%.
	Factor 6. ¿Está adecuadamente redactado (en el lenguaje de usuario) el flujo de eventos?	Métrica 7. Grado de adecuación de la descripción del flujo de eventos para un caso de uso. Umbral<10	Total de descripciones: 7 La descripción se define en el lenguaje del usuario. Se define el responsable de cada acción: 0 Representa: 0%.
	Factor 7. ¿La descripción del flujo de eventos se inicia con la descripción de una acción extrema originada por un actor o por una condición interna del sistema claramente identificable?	Métrica 8. Número de casos de uso cuya descripción incluida no inicia con una acción externa o con una condición monitoreada por el sistema. Umbral<10	Total de casos de uso:10 Número de casos de uso cuya descripción incluida no inicia con una acción externa o con una condición monitoreada por el sistema: 0 Representa: 0%.
	Factor 8. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?	Métrica 9. Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos. Umbral<10	Total de casos de uso: 10 Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Representa: 0%.
			Para un incumplimiento de 0%

Capítulo 3: Validación

Correctitud	Factor 9. ¿Representa el caso de uso requisitos comprensibles por el usuario?	Métrica 10. Número de casos de uso en que los requisitos representados no son comprensibles por el usuario. Umbral<10	Total de casos de uso: 10 Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0 Representa: 0%.
	Factor 10. ¿Las interacciones definidas describen la funcionalidad requerida del sistema?	Métrica 11. Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema. Umbral<10	Total de casos de uso:10 Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema: 0 Representa: 0%.
	Factor 11. ¿Las interacciones introducen mejoras al proceso actual?	Métrica 12. Número de casos de uso que deben ser modificados para mejorar el proceso actual. Umbral<10	Total de casos de uso:10 Número de casos de uso que deben ser modificados para mejorar el proceso actual: 1 Representa: 10 %.
			Para un incumplimiento: 3.33%
Complejidad	Factor 12. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	Métrica 13. Número de elementos del diagrama que requieren reubicación. Umbral<30	Total de elementos del diagrama:13 Número de elementos del diagrama que requieren reubicación: 1 Representa: 7.7%.

Capítulo 3: Validación

			Para un incumplimiento de: 7.7 %.
--	--	--	-----------------------------------

Tabla 3: Métricas para validar Diagrama de Caso de Uso del Sistema.

Segunda Revisión.

Atributo	Factor	Métrica Asociada	Valor
Compleitud	Factor 1. ¿Han sido definidos todos los roles relevantes del usuario encargado de generar/modificar o consultar información?	Métrica 1. Número de roles relevantes omitidos. Umbral < 10	Total de roles: 3 Número de roles relevantes omitidos: 0 Se presenta un 100%.
	Factor 2. ¿Se presenta una descripción resumida de todos los casos de uso?	Métrica 2. Número de casos de uso que no tiene descripción resumida. Umbral < 10	Total de CU: 10 Número de casos de uso que no tiene descripción resumida: 0 Representa: 0%.
	Factor 3. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	Métrica 3. Número de requisitos omitidos por casos de uso. Umbral < 10	Total de requisitos funcionales: 14 Número de requisitos omitidos por casos de uso: 0 Representa: 0%.
		Métrica 4. Número de casos de uso que tienen requisitos	Total de CU: 10 Número de casos de uso que tienen requisitos

Capítulo 3: Validación

		omitidos. Umbral < 10	omitidos: 0 Se presenta un 100%.
	Factor 4. ¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia en (crítico, importantes y útiles)?	Métrica 5. Número de casos que no han sido clasificados. Umbral < 10	Total de CU: 10 Número de casos que no han sido clasificados: 0 Representa: 0%.
			Para un incumplimiento de: 0 %.
Consistencia	Factor 5. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	Métrica 6. Número de casos de uso que tienen un nombre incorrecto. Umbral < 10	Total de CU: 10 Número de casos de uso que tienen un nombre incorrecto: 0 Representa: 0%.
	Factor 6. ¿Está adecuadamente redactado (en el lenguaje de usuario) el flujo de eventos?	Métrica 7. Grado de adecuación de la descripción del flujo de eventos para un caso de uso. Umbral < 10	Total de descripciones: 7 La descripción se define en el lenguaje del usuario. Se define el responsable de cada acción: 0 Representa: 0%.
	Factor 7. ¿La descripción del flujo de eventos se inicia con la descripción de una	Métrica 8. Número de casos de uso cuya descripción incluida no inicia con una acción	Total de CU: 10 Número de casos de uso cuya descripción incluida no inicia con una acción

Capítulo 3: Validación

	acción extrema originada por un actor o por una condición interna del sistema claramente identificable?	externa o con una condición monitoreada por el sistema. Umbral < 10	externa o con una condición monitoreada por el sistema: 0 Representa: 0%.
	Factor 8. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?	Métrica 9. Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos. Umbral < 10	Total de CU: 10 Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Representa: 0%.
			Para un incumplimiento de: 0 %.
Correctitud	Factor 9. ¿Representa el caso de uso requisitos comprensibles por el usuario?	Métrica 10. Número de casos de uso en que los requisitos representados no son comprensibles por el usuario. Umbral < 10	Total de CU: 10 Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0 Representa: 0%.
	Factor 10. ¿Las interacciones definidas describen la funcionalidad requerida del sistema?	Métrica 11. Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema. Umbral < 10	Total de CU: 10 Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema: 0

Capítulo 3: Validación

			Representa: 0%.
	Factor 11. ¿Las interacciones introducen mejoras al proceso actual?	Métrica 12. Número de casos de uso que deben ser modificados para mejorar el proceso actual. Umbral < 10	Total de CU :10 Número de casos de uso que deben ser modificados para mejorar el proceso actual: 0 Representa: 0%.
			Para un incumplimiento de: 0 %.
Complejidad	Factor 12. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	Métrica 13. Número de elementos del diagrama que requieren reubicación. Umbral < 30	Total de elementos de casos de uso: 13 Número de elementos del diagrama que requieren reubicación: 0 Representa: 0%.
			Para un incumplimiento de: 0 %.

Tabla 4: Métricas para validar Diagrama de Caso de Uso del Sistema.

Gráfica de resultados de la Métrica en las dos revisiones.

Capítulo 3: Validación

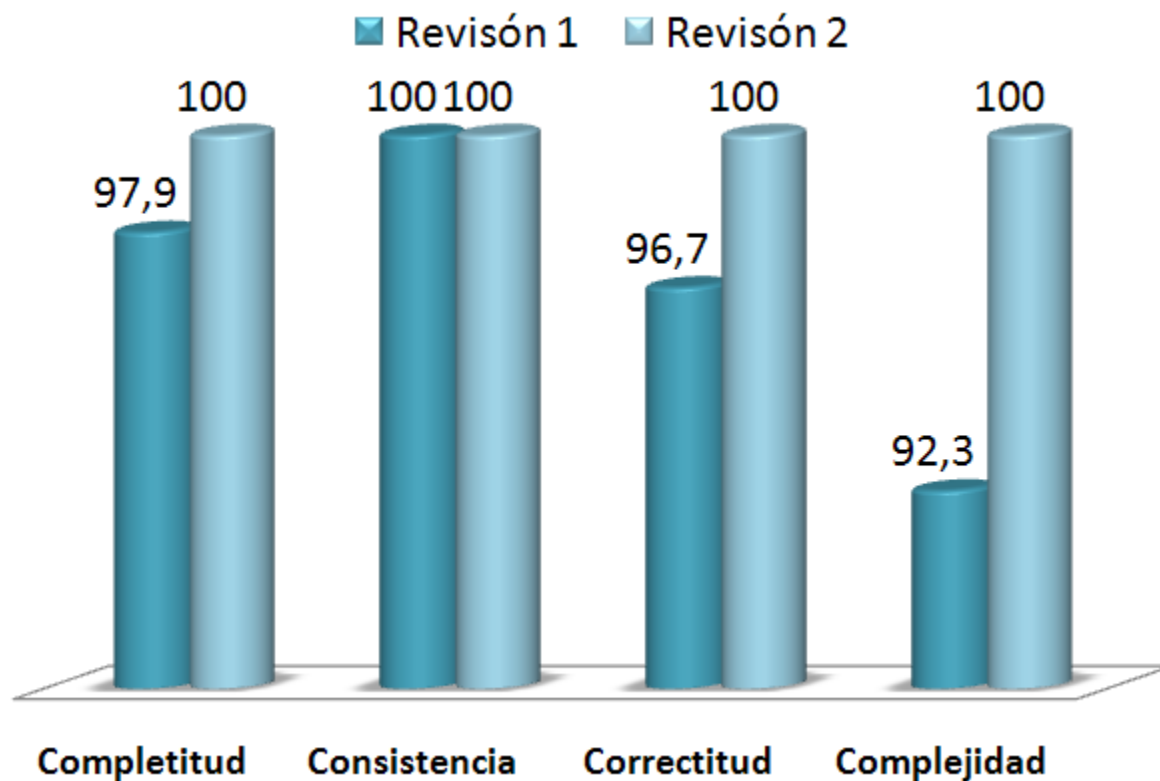


Figura 8: Resultados de la Métrica.

En la primera evaluación realizada, el DCUS alcanzó una calificación de 97.34% de funcionalidad, la contribución de cada atributo a la calidad total fue: Compleitud 97.9%, Consistencia 100%, Correctitud 96.7%, Complejidad 92.3%. Mientras que en la segunda, el DCUS alcanzó una calificación de 100% de funcionalidad, la contribución de cada atributo a la calidad total fue: Compleitud 100%, Consistencia 100%, Correctitud 100%, Complejidad 100%.

Luego de la aplicación del modelo de Métricas Orientada a Objeto al DCUS en una segunda iteración, se demostró que el mismo posee la calidad requerida para el futuro diseño del sistema debido a que: cumple con todos los requisitos identificados a través de los casos de uso, los cuales presentan una descripción detallada con todas las acciones del flujo de eventos. En las descripciones de los casos de uso complejos están bien definidas las acciones que corresponden al flujo básico de eventos y a los flujos alternos. Para

Capítulo 3: Validación

lograr una mejor comprensión del DCUS, todos los casos de uso se nombraron con una expresión verbal que describe una funcionalidad relevante para el usuario, representando así una interacción observable para un actor del sistema. Los elementos dentro del diagrama están correctamente ubicados lo que facilita su interpretación.

3.5 Matriz de trazabilidad

La Matriz de trazabilidad de requisitos es una técnica que se utiliza para ver si los casos de uso engloban todos los requisitos funcionales del sistema. Para la verificación del mismo se creó una tabla con los requisitos funcionales y los casos de uso, marcando con una x la intersección de cada uno de ellos teniendo en cuenta la especificación de requisitos.

	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10
RF1						X				
RF2	X									
RF3		X								
RF4			X							
RF5								X		
RF5.1								X		
RF5.2								X		
RF5.3								X		
RF5.4								X		
RF6				X						
RF7									X	

Capítulo 3: Validación

RF8							X			
RF9					X					
RF10										X

Tabla 5: Matriz de trazabilidad de requisitos

CU1: Modificar Contenido.

CU2: Deshabilitar Contenido.

CU3: Habilitar Contenido.

CU4: Consultar Evaluación.

CU5: Solicitar Ayuda. **(No Crítico)**

CU6: Visualizar Contenido. **(No Crítico)**

CU7: Rechazar Ejercicio.

CU8: Estudiar Contenido.

CU9: Revalorizar Evaluación.

CU10: Abandonar Tema. **(No Crítico)**

Resultados de la Métrica:

Realizada la Matriz de trazabilidad se obtuvo como resultado que los requisitos funcionales tienen relación con al menos un caso de uso, lo cual demuestra que el diagrama de casos de uso del sistema cumple con todos los requisitos descritos en la especificación de los mismos.

3.6 Validación del Diseño.

3.6.1 Métrica Tamaño operacional de clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

Atributo de Calidad.	Modo en que lo afecta.
Responsabilidad.	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.

Capítulo 3: Validación

Complejidad de Implementación.	Un aumento del TOC implica un aumento en la complejidad de implementación de la clase.
Reutilización.	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Atributo	Categoría	Criterio
Responsabilidad (Rp).	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio
Complejidad de Implementación (CI).	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio
Reutilización (Rt).	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio

Resultados del instrumento de evaluación de la Métrica Tamaño Operacional de clase (TOC):

Total de clases	Promedio de procedimientos
12	4.46

Tabla 6: Cantidad de clases y promedio de procedimientos.

Criterios	Cantidad de clases	Promedio
Entre 1 y 5 procedimientos	9	69.23076923
Entre 6 y 10 procedimientos	3	23.07692308
Entre 11 y 15 procedimientos	1	7.692307692

Capítulo 3: Validación

Entre 16 y 20 procedimientos	0	0
Entre 21 y 25 procedimientos	0	0
Más de 26 procedimientos	0	0
Total	13	100

Tabla 7: Cantidad de procedimientos por clases.

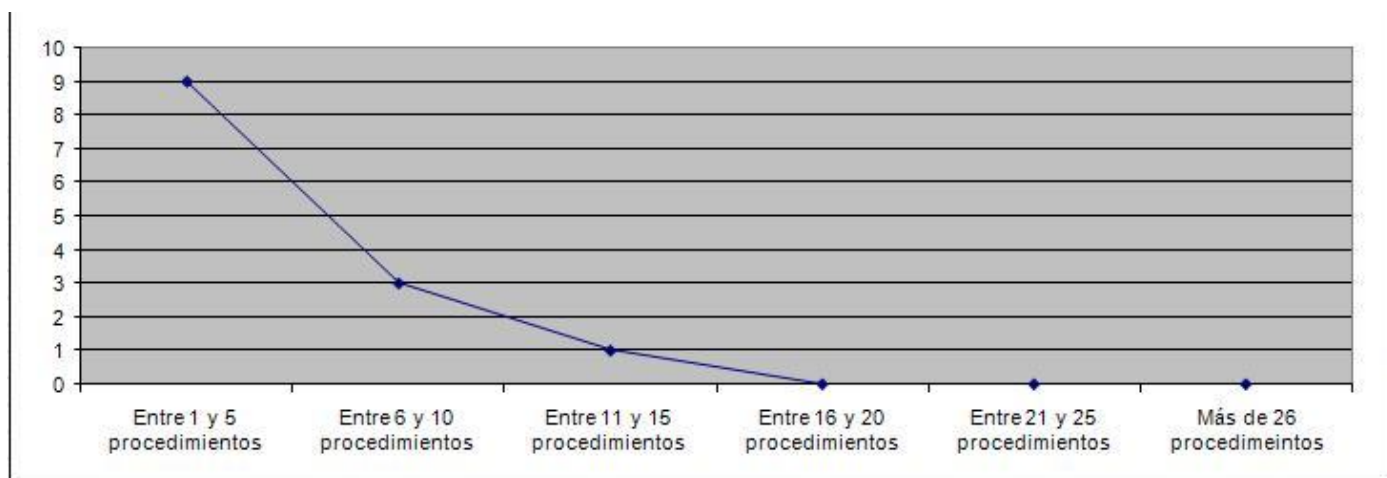
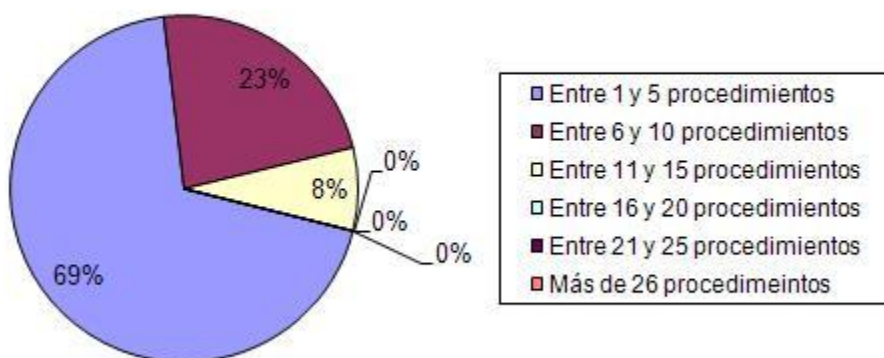


Figura 9: Gráfica de los resultados de la evaluación de la métrica TOC agrupados por las tendencias de los Valores.



Capítulo 3: Validación

Figura 10: Representación en % de los resultados obtenidos en el instrumento agrupados en intervalos definidos.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en cada uno de los atributos:

Responsabilidad	Cantidad de clases	Promedio
Bajo	9	75
Medio	3	25
Alto	1	8.333333333

Tabla 8: Atributo Responsabilidad.



Figura 11: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

Complejidad	Cantidad de clases	Promedio
Baja	9	75
Media	3	25
Alta	1	8.333333333

Tabla 9: Atributo Complejidad.

Capítulo 3: Validación



Figura 12: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad.

Reutilización	Cantidad de clases	Promedio
Bajo	9	75
Medio	3	25
Alto	1	8.333333333

Tabla 10: Atributo Reutilización.

Capítulo 3: Validación

Reutilización

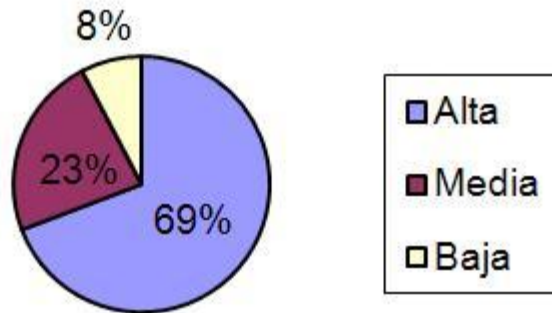


Figura 13: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Se concluye que el diseño propuesto se encuentra entre los límites aceptables de calidad, teniendo en cuenta que la mayoría de las clases (69%) posee menos cantidad de operaciones que la media registrada en las mediciones y los atributos de calidad se presentan un nivel medio satisfactorio en el 69% de las clases; se puede observar cómo se fomenta la Reutilización y cómo están reducidas en menor grado la Responsabilidad y la Complejidad de implementación.

3.6.2 Métrica de Relaciones entre clases (RC)

Atributo de Calidad.	Modo en que lo afecta.
Responsabilidad.	Un aumento del RC implica un aumento de la responsabilidad asignada a la clase.
Complejidad del mantenimiento.	Un aumento del RC implica un aumento en la complejidad del mantenimiento de la clase.
Reutilización.	Un aumento del RC implica una disminución del grado de reutilización de la clase.

Capítulo 3: Validación

Cantidad de pruebas.	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
-----------------------------	--

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Atributo.	Categoría.	Criterio.
Acoplamiento.	Ninguno.	0
	Bajo.	1
	Medio.	2
	Alto.	>2
Complejidad de mantenimiento.	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio
Reutilización.	Baja.	$> 2 \times$ Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	\leq Promedio
Cantidad de pruebas.	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio

Resultados del instrumento de evaluación de la Métrica Relaciones entre clases (RC):

Total de clases	Promedio de asociaciones de uso
13	1.3

Tabla 11: Cantidad de clases y promedio de asociaciones de uso.

Capítulo 3: Validación

Criterio	Categoría	Cantidad de clases	Promedio
0 dependencias	Muy Bueno	3	23.07692308
1 dependencias	Bueno	5	38.46153846
2 dependencias	Regular	3	23.07692308
3 dependencias	Malo	2	15.38461538
>3 dependencias	Muy Malo	0	0
Total		13	100

Tabla 12: Cantidad de dependencias por clasificación.

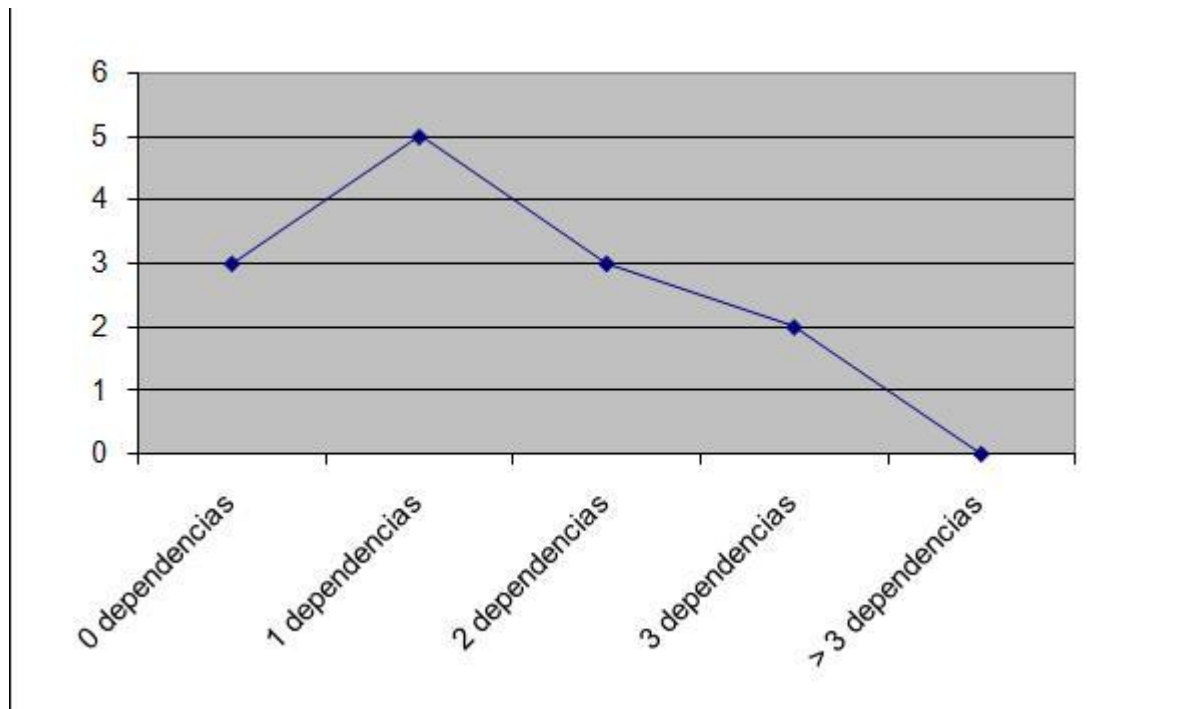


Figura 14: Gráfica de los resultados de la evaluación de la Métrica RC agrupados por las tendencias de los valores.

Capítulo 3: Validación

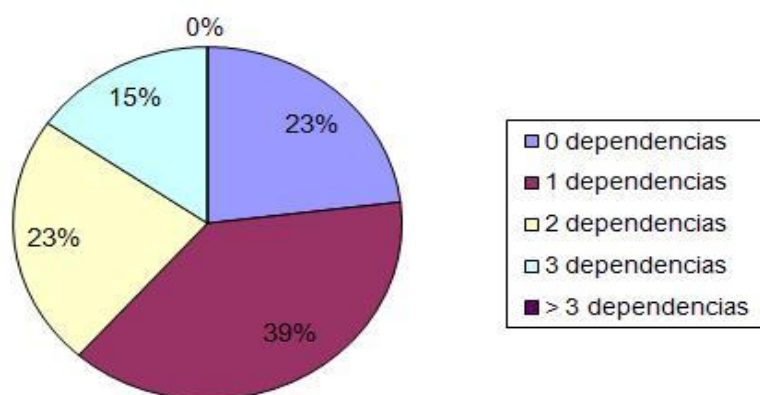


Figura 15: Representación en % de los resultados obtenidos en el instrumento agrupados en intervalos definidos.

Representación de la incidencia de los resultados de la evaluación de la Métrica RC en cada uno de los atributos:

Acoplamiento	Cantidad de clases	Promedio
Ninguno	3	23.07692308
Bajo	8	61.53846154
Medio	3	23.07692308
Alto	2	15.38461538

Tabla 13: Atributo Acoplamiento.

Capítulo 3: Validación



Figura 16: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Acoplamiento.

Complejidad de Mantenimiento	Cantidad de clases	Promedio
Bajo	8	61.53846154
Medio	2	14.28571429
Alto	1	7.142857143

Tabla 14: Atributo Complejidad de Mantenimiento.

Capítulo 3: Validación

Complejidad de Mantenimiento

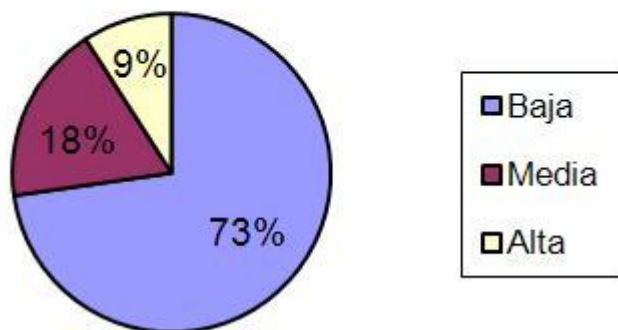


Figura 17: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Mantenimiento.

Cantidad de Pruebas	Cantidad de clases	Promedio
Bajo	8	61.53846154
Medio	2	15.38461538
Alto	2	15.38461538

Tabla 15: Atributo Cantidad de pruebas.

Capítulo 3: Validación

Cantidad de Pruebas

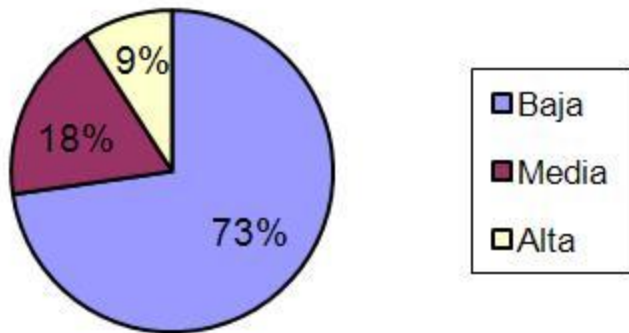
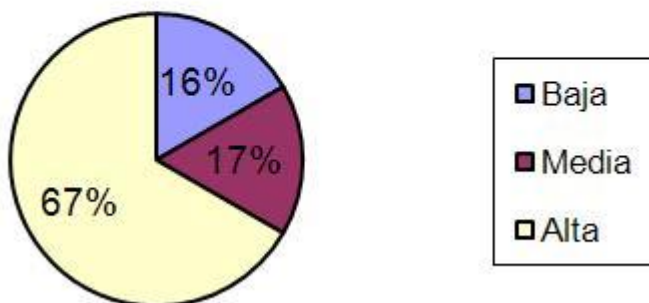


Figura 18: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Mantenimiento.

Reutilización	Cantidad de clases	Promedio
Bajo	2	15.38461538
Medio	2	15.38461538
Alto	8	61.53846154

Tabla 16: Atributo Reutilización.

Reutilización



Capítulo 3: Validación

Figura 19: Representación de la incidencia de los resultados de la evaluación de la Métrica RC en el atributo Reutilización.

Se concluye que el diseño propuesto está entre los límites aceptables de calidad, teniendo en cuenta que la mitad de las clases (51%) poseen menos de 3 dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio, el grado de acoplamiento es mínimo, la Complejidad de Mantenimiento, la Cantidad de Pruebas y la Reutilización para un 73%, 73%, 67% respectivamente, se comportan favorablemente.

3.7 Conclusiones del capítulo.

Se aplicó en este capítulo las Métricas a los requisitos y a los casos de uso del sistema para comprobar la calidad que presentaban los mismos, además de la confección de la Matriz de trazabilidad para detectar si existía algún requisito funcional que no se relacionara con al menos uno de los casos de uso del sistema. Las Métricas para validar el diseño que se pusieron en práctica fueron: Tamaño Operacional de Clases (TOC) y Responsabilidad entre Clases (RC) para verificar la calidad que presentan estos aspectos.

Conclusiones

Conclusiones

- Con el estudio de los diferentes Laboratorios Virtuales y los sistemas de aprendizaje existentes se logró identificar las principales necesidades del sistema en cuestión.
- Mediante un profundo análisis de las herramientas existentes como Visual Paradigm y CompendiumLD se logró obtener los artefactos generados en el diseño del módulo de Exclusión Mutua y Sincronización.
- La aplicación de Métricas a los requisitos funcionales y a los casos de uso del sistema permitió saber la calidad que presentaban los mismos y de este modo validar los resultados obtenidos.

Recomendaciones

Recomendaciones

Se recomienda para el seguimiento de la investigación:

- Realizar la implementación del Laboratorio Virtual de Sistemas Operativos, basándose en la modelación realizada en el trabajo.
- Continuar perfeccionando el modelado del sistema mediante la actualización de los cambios que sean necesarios durante las etapas de implementación y pruebas.

Referencias Bibliográficas

Referencias Bibliográficas

1. **Ricardo. Febe Ángel Ciudad.** ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI. Ciudad de la Habana: s.n. 2007.
2. **Pérez. Dairelis Pérez González y Grenia Hernández.** Guía para realizar Ingeniería de Requisitos a la línea de productos informáticos que utilizan tecnología multimedia. 2010.
3. **Cabrera. Carlos Rafael Cabrera y Manuel.** Implementación del Módulo de Verificaciones Fiscales en el Sistema de la Fiscalía Militar. 2010.
4. Metodología para el desarrollo de material educativo orientado a objetos de aprendizaje. s.l. : CATED, 2007
5. F. Aznar Gregori. Los Mapas conceptuales y su aportación al proceso de aprendizaje en la asignatura MFAC.
6. Compendium Institute. [Online] [Cited: Mayo 3, 2011.] <http://compendium.open.ac.uk/institute/>.
7. monografias.com. [Online] [Cited: Abril 23, 2011.] <http://www.monografias.com/trabajos65/objetos-aprendizaje/objetos-aprendizaje.shtml>.
8. **Sommerville,** Ian. 2005. Ingeniería de Software. 2005.
9. **Urrutia. Lianni Yadira Figueredo y Misael Rodriguez.** Simulador para la asignatura Sistemas Operativos. La Habana: s.n., 2010.
10. Profundización del flujo de trabajo de requerimientos. La Habana: s.n. 2008-2009. Asignatura: Ingeniería de Software I.
11. **Garcia. Joaquin.** .Patrones de diseño. Diseño de software orientado a objeto. 27 de mayo 2005.
12. **Asmoz.** [Online] [Cited: Abril 12, 2011] http://www.asmoz.org/index.php?option=com_content&task=view&id=55&Itemid=80.
13. **Pressman.** Ingeniería del Software. Un enfoque práctico 6ta Edición. 2005.

Referencias Bibliográficas

14. EcuRed [Online] [Cited: Abril 12, 2011]
http://www.ecured.cu/index.php/Patrones_de_dise%C3%B1o_y_arquitectura

Bibliografía

Bibliografía

Urrutia., Lianni Yadira Figueredo y Misael Rodriguez. Simulador para la asignatura Sistemas Operativos. La Habana: s.n., 2010. .

Ricardo, Febe Angel Ciudad. ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI. La Habana: s.n., 2007.

[Online] <http://www.upv.es/upl/U0471649.pdf>.

Maestros del web. [Online] [Cited: Abril 15, 2011.] <http://www.maestrosdelweb.com/editorial/multime/>.

Pérez, Dairelis Pérez González y Grenia Hernández. Guía para realizar Ingeniería de Requisitos a la línea de productos informáticos que utilizan tecnología multimedia. La Habana: s.n., 2010.

Juan C. Dürsteler. **InfoVis.net** [Online] [Cited: Mayo 12, 2011.] <http://www.infovis.net/printMag.php?num=141&lang=1>.

Pressman. Ingeniería del Software. Un enfoque práctico 6ta Edición. 2005.