

**UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**DESARROLLO DE UN SOFTWARE EDUCATIVO PARA
FACILITAR EL PROCESO ENSEÑANZA-APRENDIZAJE DE LA
ASIGNATURA MODELOS DE OPERACIONES I, DICTADA EN
EL DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS DE LA
UNIVERSIDAD DE ORIENTE**

REALIZADO POR:

María Alejandra Salazar Rengel

Reyna Isabel Cadenillas Espinoza

**Trabajo de grado presentado en la Universidad de Oriente como requisito
parcial para optar al título de**

**INGENIERO EN COMPUTACIÓN E INGENIERO DE SISTEMAS,
RESPECTIVAMENTE**

Barcelona, Julio de 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UN SOFTWARE EDUCATIVO PARA
FACILITAR EL PROCESO ENSEÑANZA-APRENDIZAJE DE LA
ASIGNATURA MODELOS DE OPERACIONES I, DICTADA EN
EL DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS DE LA
UNIVERSIDAD DE ORIENTE**

ASESORADO POR:

Ing. Zulirais García

Barcelona, Julio de 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UN SOFTWARE EDUCATIVO PARA
FACILITAR EL PROCESO ENSEÑANZA-APRENDIZAJE DE LA
ASIGNATURA MODELOS DE OPERACIONES I, DICTADA EN
EL DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS DE LA
UNIVERSIDAD DE ORIENTE**

JURADO CALIFICADOR:

Ing. Zulirais García

Asesor Académico

Ing. Mónica Saettone

Jurado Principal

Ing. Pedro Dorta

Jurado Principal

Barcelona, Julio de 2009

RESOLUCIÓN

De acuerdo al artículo 44 del reglamento de trabajos de grado de la Universidad de Oriente:

“Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y solo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participara al Consejo Universitario”

DEDICATORIA

Dedicado a mi familia, especialmente, a mis padres Francisco Salazar y Neri Rengel, quienes me han apoyado incondicionalmente a lo largo de mi carrera y de mi vida. Sin ustedes este logro no hubiese sido posible. Este título es para ustedes.

Maria Alejandra. Salazar

DEDICATORIA

Solo a ti madre, un ser humano increíblemente fuerte, tanta fortaleza permitio agrandar mis esfuerzos. Solo tú y yo sabemos todas las adversidades que hemos superado. Porque me diste la vida y en mi vida siempre has estado. No existe otra persona a quién yo le dedique mi esfuerzo y trabajo.

Reyna I. Cadenillas E.

AGRADECIMIENTOS

Agradezco a dios por darme la fuerza, la paciencia y la perseverancia, necesaria para culminar con éxito este logro en mi vida, por observarme día a día, mantenerme firme en mis propositos y no permitirme renunciar.

Agradezco, inmensamente a mis padres, por estar siempre conmigo, ayudarme, cuidarme, entenderme, y darme el apoyo necesario, para culminar con éxito mi carrera.

Agradezco a mis hermanos Fran y Daniel, por ayudarme cuando lo necesité (especialmente cuando me fracture el tobillo). Siempre estaban ahí para hacerme algún favor, por eso Gracias.

Gracias a mi compañera de tesis (Reyna), primero por aceptar participar en este proyecto conmigo, y segundo, por tenerme paciencia, entenderme, compenetrarse en la ejecución de la tesis y ayudarme a finalizar este trabajo de grado exitosamente.

Gracias a la profesora Zulirais por asesorarme y estar disponible para mi, ofreciendome su ayuda y conocimientos para el desarrollo de este trabajo.

Agredezco, especialmente, a Jean Carlos Diaz, por ayudarme o basicamente enseñarme php, creo que sin tu ayuda hubiese sufrido más de lo necesario. Muchas gracias.

Agradezco a todos mis amigos y compañeros de clase, que estuvieron conmigo a lo largo de mi carrera, me brindaron una mano amiga y su incondicional apoyo.

Agradezco a todos aquellos que, de alguna manera, contribuyeron en la culminación de mi carrera y/o en la elaboración de mi tesis.

Finalmente, agradezco a los profesores de la Universidad de Oriente por ofrecerme día a día sus conocimientos, y hacerme la ingeniero que soy hoy en día. Muchas gracias.

Maria Alejandra Salazar

AGRADECIMIENTOS

Primero quiero dar gracias a dios por brindarme la oportunidad de poder redactar esta hoja de agradecimientos y porque existen muchas personas a quien debo agradecer.

Mi mamá, Amancia Espinoza. Mami gracias por todo el esfuerzo que haces por mí, eres indispensable para este logro en mi vida, tu apoyo me ayudo a sobresalir en los momentos malos y a seguir creciendo en los momentos buenos. Un millón de Gracias mami. TE AMO.

A mis hermanos, Julián Cadenillas, quien siempre actuó como un padre para mi, aquí, en tierras venezolanas y a quien agradezco el estar pendiente de mi y su gran apoyo en todo momento, Gonzalo Cadenillas, quien actuó como un padre para mi en tierras peruanas, fueron inolvidables los momentos a tu lado hermano, y a mi hermana mayor Clotilde Cadenillas, por estar siempre ahí, a mi lado.

Ciudad Bolívar lugar donde conocí a grandes personas: Glendismar, es increíble el tiempo que llevamos compartiendo, y los que vienen amiga. Beba, guao! increíbles momentos que siempre recordamos. Maria Alejandra, realmente fue un placer para mí haberte conocido amiga. Henry, un gran amigo, Neiry, una excelente chica. Es bueno saber que siempre conté con ustedes en las buenas y en las malas, a pesar de la distancia nunca sentí la ausencia, los quiero un montón amigos. Gracias

Aunque parezca fuera de lugar doy gracias al comedor de la Universidad de Oriente, lugar donde compartí con muchas personas queridas y lugar que conocí gracias a mi amigo Orwill Rojas quien fue como una guía para mí, cuando Anzoátegui me era completamente extraño, también, me presentara a Daniel Aguilera

y Francisco Molero (sus grandes amigos). Chicos son excelentes personas y en realidad les tengo mucho que agradecer.

Al hablar del comedor, por supuesto, tengo que mencionar a David Romero, un amigo fuera de lo común, un amigo chévere, quien junto a su hermano Rodrigo Romero, compartimos y disfrutamos muchos momentos, gracias chicos.

Quiero agradecer a Leonerso Villazana, quien durante mucho tiempo me brindo su apoyo y me ayudo en los momentos más difíciles, también le agradezco haberme incluido en el grupo de trabajo de La Sala de Lectura de Computación y Sistemas, porque ahí conocí a magníficas personas: Hernán, Andreina, Carolina, Rosmara, Nohely y Romilé, muchas gracias chicos, gracias por todo.

Existen esas personas de las que uno no se entera de cómo llegaron a nuestras vidas pero al percatarse de que están ahí no se quieren perder en lo absoluto y se agradece su amistad: Enrique Marcano, Nathaly Gómez y Gabriela Milano.

Por supuesto que tengo que mencionar a mis compañeros de residencia, por que juegan el rol de familia sustituta, les doy gracias porque realmente fueron una buena familia: Vanesa Velásquez, Silvia Rodríguez, Magiore Prieto, Livinel Guaita, Renier Ruiz y Osmael Duarte

El haber desarrollo este trabajo me permitió conocer muchas cosas nuevas, fue una experiencia realmente batalladora, el cual ha sido superado junto a mi compañera Maria A. Salazar, a quién agradezco haberme brindado esa oportunidad y confiar en mi como compañera de tesis.

A La profesora Zulirais García, a quien agradezco haber aceptado formar parte de este trabajo como asesora. Profesora, sin sus consejos y guías no se hubiera podido

desarrollar de manera exitosa este trabajo, gracias. Mil bendiciones para usted y su familia. Es usted una excelente persona.

¡Ah creyeron que me iba a olvidar de ustedes! Pero no, imposible olvidarlos, no podría olvidarlos. Fíjense que también tengo muchas cosas que agradecerles, les agradezco la amista, la alegría, el compartir, el cariño, el apoyo, escucharme y hacerme sentir que siempre cuento con ustedes. Gracias: Héctor Sandoval, Deivys Brito, Adrián Carvajal, Tania Martínez, Cesar Pérez y Wladimir Blanco. Definitivamente fue buena decisión ver matemáticas IV en intensivo, porque, esa decisión me brindo la oportunidad de conocerlos. Los Quiero Mucho

Reyna I. Cadenillas E.

RESUMEN

El trabajo de grado presentado, consiste en el desarrollo de un software educativo que dará apoyo al sistema educativo empleado en el departamento de computación y sistemas de la Universidad de Oriente, facilitando el proceso de enseñanza-aprendizaje de la asignatura Modelos de Operaciones I, dictada como cátedra reglamentaria para ingeniería de sistemas y como electiva técnica para ingeniería en computación. Con la realización del software educativo de Modelos de Operaciones I (SEMOI) se persigue el desarrollo de un proyecto completo que ofrezca una alternativa tecnológica e interactiva para el sistema de educación tradicional y que, a su vez, permita reforzar conocimientos y facilite la labor docente a través del uso de tecnología multimedia y la World Wide Web. En el SEMOI se expondrá de manera didáctica, clara y detallada cada uno de los temas que forman parte del contenido de la asignatura, utilizando diversas herramientas audiovisuales como imágenes estáticas y dinámicas, juegos y animaciones, además, se realizan evaluaciones al final de cada capítulo que permitirán, tanto a profesores como a estudiantes, determinar si el aprendizaje se está llevando a cabo de una manera efectiva. Para su creación se toma como base el Proceso Unificado de Desarrollo de Software que, junto al Lenguaje Unificado de Modelado, permite la planificación y estructuración de la aplicación. Aunado a esto, se emplea la metodología para la creación de software educativo propuesta por Álvaro Galvis, que permite completar el modelado del sistema. Para el diseño y construcción del modelo educativo se usa Macromedia Flash MX 2004, que permite representar el contenido de la asignatura mediante imágenes, texto y animaciones. Dreamweaver MX 2004 se usa para la implementación de la aplicación en una página HTML y/o PHP que será mostrada en Internet. Finalmente, se emplea MySQL para la implementación de la base de datos que sirve de apoyo para la ejecución de las evaluaciones y soporte de información de usuarios.

CONTENIDO

	Pag
RESOLUCIÓN.....	IV
DEDICATORIA.....	V
DEDICATORIA.....	VI
AGRADECIMIENTOS.....	VIII
AGRADECIMIENTOS.....	X
RESUMEN	XIII
CONTENIDO.....	XIV
CAPÍTULO I: INTRODUCCIÓN	18
1.1 PLANTEAMIENTO DEL PROBLEMA	18
1.2 OBJETIVOS.....	22
1.2.1 Objetivo General	22
1.2.2 Objetivos Específicos	22
CAPÍTULO II: MARCO TEÓRICO.....	23
2.1. ANTECEDENTES.....	23
2.2 FUNDAMENTOS TEORICOS.....	26
2.2.1 Proceso Enseñanza-Aprendizaje.....	26
2.2.2 Recursos Didáctico-Pedagógicos	26
2.2.3 Software Educativo	27
2.2.3.1 Características Esenciales del Software Educativo.....	28
2.2.3.2 Estructuras Básicas del Software Educativo	29
2.2.3.3 Clasificación del Software Educativo.....	31
2.2.3.4 Funciones del Software Educativo	33
2.2.4 Materiales Educativos Computarizados	35
2.2.5 Ingeniería de software educativo (ISE)	36
2.2.5.1 Metodología para el Desarrollo de Software Educativo	36
2.2.5.2 La Ingeniería de Software y el Paradigma Orientado a Objetos (O.O.).....	37
2.2.5.3 Metodología ISE O.O.	39

2.2.6 Multimedia	48
2.2.7 Proceso Unificado De Desarrollo De Software	50
2.2.7.1 Aspectos distintivos del Proceso Unificado.....	50
2.2.7.2 Fases del Proceso Unificado de Desarrollo de Software.....	52
2.2.8 Lenguaje Unificado De Modelado (UML)	55
2.2.8.1 Diagramas UML.....	55
2.2.8.2 UML para la Web.....	62
2.2.8.3 WAE: Web Application Extensión.....	63
2.2.9 Herramientas De Desarrollo	67
2.2.9.1 Herramienta para creación de animaciones: FLASH.....	68
2.2.9.2 Lenguaje de creación de páginas Web: HTML	68
2.2.9.3 Lenguaje de Acceso a Base de Datos: SQL.....	69
2.2.9.4 Lenguaje de Programación para Páginas Web: PHP	69
2.2.10 Contenido Programático de la Asignatura Modelos de Operaciones I.....	70
CAPÍTULO III: FASE DE INICIO	72
3.1 INTRODUCCIÓN.....	72
3.2 FLUJO DE TRABAJO: REQUISITOS	73
3.2.1 Identificación de Requisitos Candidatos	73
3.2.2 Contexto del Sistema.....	74
3.2.3 Requisitos Funcionales.....	76
3.2.3.1 Análisis de Necesidades Educativas de La Población Objetivo.....	76
3.2.3.2 Especificación de los Requerimientos	96
3.2.3.3 Identificación de Riesgos.....	97
3.2.3.4 Características del Proceso de Enseñanza - Aprendizaje	98
3.2.3.5 Característica de la Población Objetivo.....	99
3.2.3.6 Representación de los Requisitos como Casos De Uso	100
3.2.4 Requisitos No Funcionales	104
3.2.4.1 Requisitos Adicionales	105
3.3 FLUJO DE TRABAJO: ANÁLISIS	107
3.3.1 Identificación de Paquetes de Análisis.	108
3.3.2 Identificación de Clases de Análisis.	109
3.3.3 Realización de Casos de Uso-Análisis	112
3.3.3.1 Diagrama de Clases de Análisis	112
3.3.3.2 Diagrama de Interacción.....	113
3.3.3.3 Descripción de la Aplicación.....	115
3.3.3.4 Descripción de los Escenarios de Interacción.....	116

3.4 EVALUACIÓN DE LA FASE DE INICIO	117
3.5 PLANIFICACIÓN DE LA FASE DE ELABORACIÓN	117
CAPÍTULO IV: FASE DE ELABORACIÓN	119
4.1 INTRODUCCIÓN.....	119
4.2 FLUJO DE TRABAJO: REQUISITOS	120
4.2.1 Representación De Los Requisitos Como Casos De Uso.....	120
4.3 FLUJO DE TRABAJO: ANÁLISIS	124
4.3.1 Identificación de Paquetes de Análisis	125
4.3.2 Identificación de Clases de Análisis	126
4.3.3 Realización de Casos de Uso-Análisis	128
4.3.3.1 Diagramas de Clases de Análisis	128
4.3.3.2 Diagramas de Interacción	130
4.4 FLUJO DE TRABAJO: DISEÑO.....	136
4.4.1 Diseño de la Interfaz.....	137
4.4.1.1 Diseño Educativo del SEMOI	137
4.4.1.2 Diseño Comunicacional.....	160
4.4.1.3 Diseño Computacional	165
4.4.2 Diseño de la Base de Datos	175
4.5 FLUJO DE TRABAJO: IMPLEMENTACIÓN	178
4.5.1 Modelo de Implementación	179
4.5.2 Implementación de las Clases.....	180
4.6 EVALUACIÓN DE LA FASE DE ELABORACIÓN	193
4.7 PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN	194
CAPÍTULO V: FASE DE CONSTRUCCIÓN	195
5.1 INTRODUCCIÓN.....	195
5.2 FLUJO DE TRABAJO: IMPLEMENTACIÓN	196
5.2.1 Escogencia de las Herramientas de Desarrollo.....	196
5.2.2 Escogencia del Gestor de Base de Datos	198
5.2.3 Construcción de la Base de Datos	199
5.2.4 Modelo de Implementación.....	201
5.3 FLUJO DE TRABAJO: PRUEBAS	203
5.3.1 Pruebas de Unidad.....	204
5.3.2 Pruebas de Integridad	212

5.3.2.1 Integración del caso de uso Registrar Usuario.....	213
5.3.2.2 Integración del caso de uso Navegar Unidades	233
5.4 EVALUACIÓN DE LA FASE DE CONSTRUCCIÓN	257
5.5 PLANIFICACIÓN DE LA FASE DE TRANSICIÓN	258
CAPÍTULO VI: FASE DE TRANSICIÓN.....	259
6.1 INTRODUCCIÓN.....	259
6.2 PLANIFICACIÓN DE LA INSTALACIÓN DE LA VERSIÓN BETA	259
6.3 DOCUMENTACIÓN	260
6.4 EVALUACIÓN DE LA FASE DE TRANSICIÓN	261
CONCLUSIONES.....	262
RECOMENDACIONES.....	264
BIBLIOGRAFÍA	266
ANEXOS.....	¡ERROR! MARCADOR NO DEFINIDO.

CAPÍTULO I: INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

La Investigación de Operaciones, es una rama de las Matemáticas que se fundamenta en el uso de modelos matemáticos, estadística y algoritmos con objeto de realizar un proceso de toma de decisiones. Frecuentemente, trata el estudio de complejos sistemas reales, con la finalidad de mejorar (u optimizar) el funcionamiento del mismo. La Investigación de Operaciones permite el análisis de la toma de decisiones teniendo en cuenta la escasez de recursos, para determinar cómo se pueden maximizar o minimizar los recursos.

La Investigación de Operaciones ha sido testigo de numerosas innovaciones en los últimos años, y hoy en día se ha convertido en un tema de importancia crucial en el mundo empresarial permitiendo encontrar soluciones que atiendan de la mejor manera posible a los objetivos de una organización, razón por la cual la Investigación de Operaciones se imparte en diversas universidades para la formación de ingenieros que requieran conocimientos sobre optimización de procesos de producción.

La Universidad de Oriente forma profesionales que demandan conocimientos sobre Investigación de Operaciones, por ello, específicamente en el Departamento de Computación y Sistemas se imparte la asignatura Modelos de Operaciones I (código: 071-4633), en la cual, se exponen temas como: análisis de redes PERT-CPM, teoría de juegos, programación dinámica y programación no lineal.

Actualmente, la asignatura es dictada utilizando métodos de enseñanza teóricos y prácticos expuestos en el salón de clases directamente por el Profesor, realizando evaluaciones escritas para comprobar que se está realizando correctamente el proceso

enseñanza – aprendizaje y medir los conocimientos adquiridos por cada uno de los estudiantes, es decir, verificar que el aprendizaje se está logrando. No se utilizan con frecuencia materiales audiovisuales.

Estos métodos de enseñanza tradicionales, quizás, no resulten ser los más adecuados para esta asignatura, ya que, es mayormente práctica y probablemente se requiera de materiales audiovisuales que faciliten su comprensión. Aunado a esto, se observa la existencia de estudiantes que manifiestan apatía o rechazo por la asignatura, debido al constante uso de procedimientos y expresiones matemáticas que permiten la formulación de los modelos, empleados para representar situaciones o fenómenos de la vida real. Por otra parte, algunos estudiantes pueden presentar falta de motivación. y/o tener dificultades para entender con celeridad los temas de estudio, es decir, necesitan mayor dedicación. Estos factores originan que sólo algunos estudiantes consigan aprovechar al máximo los conocimientos adquiridos.

Siguiendo con lo descrito, el presente trabajo de investigación propone el Desarrollo de un Software Educativo como apoyo al proceso de Enseñanza – Aprendizaje de la asignatura Modelos de Operaciones I en el Departamento de Computación y Sistemas de la Universidad de Oriente, que promueva el interés del estudiante en la obtención de conocimientos, que incentive y ofrezca una alternativa práctica, efectiva y atractiva al plan de educación tradicional. Se tomarán en cuenta innovaciones tecnológicas y medios audiovisuales que permitan presentar la información de forma clara y sencilla para los estudiantes.

Para ello, se tomará como referencia lo propuesto por Álvaro Galvis (1994) en “Ingeniería de Software Educativo”, en el cual, se expone la metodología y los pasos más convenientes a seguir en la elaboración de un software educativo. Además, se utilizará el Proceso Unificado de Desarrollo de Software que, mediante el lenguaje unificado de modelado (UML), permiten construir y documentar todo el sistema.

También serán de utilidad diferentes herramientas de programación, que permitan culminar de manera satisfactoria el desarrollo de la aplicación.

El software permitirá exponer de forma dinámica el contenido de la asignatura, el cual estará dividido en unidades específicas y estos, a su vez, se dividirán en lecciones manteniendo un orden secuencial y racional en su desarrollo. Se permitirá el retroceso a temas anteriores para reforzar o aclarar los conocimientos en caso de que no hayan sido comprendidos con exactitud. Además, se incluirán animaciones y grabaciones de sonido (audio) que permitan exponer gráfica y auditivamente la resolución de problemas acerca de cada tema, buscando captar y mantener la motivación del estudiante.

Se realizarán evaluaciones interactivas al final de cada capítulo, que permitan comprobar que el aprendizaje se está logrando, igualmente pueden incluirse juegos o actividades que establezcan el diálogo continuo entre el usuario y la aplicación. Cada evaluación será realizada tomando como base el contenido explicado en el capítulo específico al cual corresponde, permitiendo que el estudiante se autoevalúe y que reconozca sus deficiencias y habilidades en la materia.

Cabe destacar que el software educativo servirá de apoyo al Profesor que imparte la materia y nunca pretenderá sustituirlo, ya que, le proporcionará una herramienta tecnológica moderna, en la cual, se aprovechen al máximo las características del computador (inclusión de fotografías, animaciones, sonidos, realidad virtual, etc.). De forma similar, servirá de soporte para el autoaprendizaje del estudiante reforzando el conocimiento suministrado por el profesor, fomentando la participación activa, al poner en práctica, mediante evaluaciones y juegos interactivos, los conocimientos adquiridos, y además permitirá que el alumno se desenvuelva a un ritmo escogido por él y que sea adecuado a sus capacidades e intereses.

El software colaborará con el proceso de enseñanza de Modelos de Operaciones I, basándose en el contenido programático de la asignatura dictada en el departamento de computacion y sistemas de la Universidad de Oriente, el cual contiene las siguientes unidades.

Unidad 1: Análisis de Redes Pert-CPM.

Unidad 2: Teoría de Juegos.

Unidad 3: Programación Dinámica.

Unidad 4: Programación No Lineal.

Esta división en unidades, permitirá la creación de una aplicación que facilite y dé apoyo al proceso enseñanza-aprendizaje de la asignatura en la Universidad de Oriente. El software se diseña, específicamente, para la asignatura Modelos de Operaciones I, dictada en el 7mo semestre, como materia obligatoria para Ingeniería de Sistemas y como electiva para Ingeniería en Computación, sin embargo, podrá ser utilizado para la educación general de la Investigación de Operaciones en otras universidades o campos de trabajo, debido a que abarca en gran parte su contenido.

1.2 OBJETIVOS

1.2.1 Objetivo General

Desarrollar un Software Educativo que facilite el proceso de Enseñanza-Aprendizaje de la asignatura Modelos De Operaciones I, dictada en el Departamento de Computación y Sistemas de la Universidad de Oriente.

1.2.2 Objetivos Específicos

- Analizar las necesidades de la población objetivo, recopilando la información necesaria para el cumplimiento de los requerimientos.
- Realizar el diseño educacional del software, tomando como base, el contenido de la asignatura Modelos de Operaciones I.
- Realizar el diseño computacional, usando el lenguaje de modelado unificado (UML).
- Codificar los componentes diseñados.
- Integrar las aplicaciones codificadas.
- Realizar las pruebas de unidad e integración.

CAPÍTULO II: MARCO TEÓRICO

2.1. ANTECEDENTES

Las primeras ideas sobre software educativo aparecen en la década de los 60, tomando mayor auge con la aparición de las microcomputadoras a principios de los 80. La aplicación de los instrumentos tecnológicos no solucionó los problemas educativos porque no hubo acompañamiento de teorías acerca de la enseñanza y el aprendizaje.

En Venezuela, han sido innumerables los esfuerzos por superar las deficiencias de la educación tradicional en colegios y universidades. La incursión de la computadora en el ámbito educativo venezolano se evidenció en 1993 con el Proyecto Simón, dotando a las escuelas de laboratorios de computación con software educativos para el mejoramiento de los procesos de enseñanza y aprendizaje de las ciencias básicas. A nivel universitario, el software educativo necesita de un impulso importante para poder ser empleado y difundido.

En la Universidad de Oriente núcleo Anzoátegui, aun no se ha realizado ningún software educativo que facilite el aprendizaje de la asignatura Modelos De Operaciones I, sin embargo, se han realizado innumerables trabajos de grado donde se desarrollan software para dar apoyo en el proceso enseñanza-aprendizaje de distintas asignaturas, tanto a nivel básico como universitario, entre ellos se pueden nombrar los siguientes:

- **Abbondanza J., Rodríguez A., (2004), “Desarrollo de un Software educativo Interactivo sobre los fundamentos de Inteligencia Artificial utilizando Tecnología World Wide Web”.**

Resumen: Tesis de Grado de Ingeniería en Computación, realizada por Abonanza Juan y Rodríguez Adriana. Consiste en la elaboración de un Software de Enseñanza – Aprendizaje para la asignatura Inteligencia Artificial dictada en el Departamento de Computación y Sistemas de la Universidad de Oriente. Se guía por el Proceso Unificado de Desarrollo de Software, la metodología de Ingeniería de Software educativo, las capacidades de diseño visual de Macromedia Flash y la arquitectura ASP.Net. [1]

- **Veracierta, G., Torres, P., (2005), “Desarrollo de un Software Educativo para Facilitar la Enseñanza y Aprendizaje de la Asignatura Análisis y Diseño de Algoritmos de las Unidades I, II, III y V; Dictada en el Departamento de Computación y Sistemas de la Universidad de Oriente Núcleo Anzoátegui”.**

Resumen: Tesis de grado de Ingeniería en Computación, presentada por Veracierta Gabriela y Torres Pamela. El proyecto presentado se basa en los principios de un curso virtual que proporciona una nueva alternativa para suministrar el material de instrucción que buscará el interés del estudiante hacia el contenido del software, tomando en cuenta las teorías de aprendizaje. Para la elaboración del software se empleo el Proceso Unificado de Desarrollo de Software aunado a la metodología de Ingeniería de Software Educativo de Álvaro Galvis. [2]

- **Rodriguez, R., (2005), “Desarrollo de un software Educativo Interactivo que Sirva de Apoyo en el Proceso de Enseñanza – Aprendizaje de la Asignatura Matemática de Octavo Grado Impartida en una Unidad Educativa”.**

Resumen: Tesis de Grado de Ingeniería en Computación, realizada por Rodríguez Roxana. El trabajo consistió en diseñar y desarrollar un software

educativo, para la asignatura matemática de octavo grado impartida en una Unidad Educativa de Educación Básica que apoye y contribuya a mejorar la enseñanza de la asignatura. En la implementación del Software se usó la metodología de software Educativo de Álvaro Galvis y el Proceso Unificado de Desarrollo de Software. [3]

- **Briceño, B., Gil, Y., (2006), “Desarrollo de un Software Educativo como Herramienta de Apoyo al Proceso de Enseñanza Aprendizaje de la Cátedra Ciencias Biológicas de Noveno Grado de Educación Básica, Utilizando la Tecnología Multimedia y World Wide Web”.**

Resumen: Tesis de Grado de Ingeniería en Computación, realizada por Briceño Betsy y Gil Yrma. El trabajo consistió en el desarrollo de un software educativo que sirve como herramienta de apoyo en el proceso de enseñanza-aprendizaje de la asignatura Ciencias Biológicas de noveno grado de educación Básica. El proyecto siguió la metodología por la creación de software interactivo propuesta por Álvaro Galvis, además se utilizó el Proceso Unificado de Desarrollo de Software para modelar todo el sistema. [4]

- **Pereira M., (2006), “Desarrollo de un Software Instruccional para la Enseñanza – Aprendizaje de la Asignatura Programación I que se Imparte en el Departamento de Computación y Sistemas de la Universidad de Oriente Núcleo Anzoátegui”.**

Resumen: Tesis de Grado de Ingeniería en Computación, realizada por Pereira Marcia. Se utilizó, conjuntamente, el Proceso Unificado de Desarrollo de Software y la metodología de Software Educativo de Álvaro Galvis, para la elaboración de un software educativo interactivo basado en una aplicación Web para la asignatura Matemáticas de noveno grado de educación Básica.

Esta aplicación ayuda a los alumnos a obtener una mejor comprensión de la asignatura. [5]

2.2 FUNDAMENTOS TEORICOS

A continuación se muestra los fundamentos teóricos y una definición de términos, los cuales sirvieron de base para el desarrollo de este proyecto.

2.2.1 Proceso Enseñanza-Aprendizaje

La enseñanza es una actividad intencional, diseñada para dar lugar al aprendizaje de los alumnos.

Debido a que el término aprendizaje vale tanto para expresar una tarea como el resultado de la misma, es fácil mezclarlos y decir que la tarea de la enseñanza es lograr el resultado del aprendizaje, cuando en realidad tiene más sentido decir que "la tarea central de la enseñanza es posibilitar que el alumno realice las tareas del aprendizaje".

Entenderemos, pues, por proceso de enseñanza-aprendizaje, el sistema de comunicación intencional que se produce en un marco institucional y en el que se generan estrategias encaminadas a provocar el aprendizaje. [6].

2.2.2 Recursos Didáctico-Pedagógicos

Son las técnicas de enseñanza que el docente seleccionará para organizar sus actividades y la de los alumnos. Los recursos didáctico-pedagógicos son los elementos empleados por el docente para organizar sus actividades y la de los

alumnos y para facilitar y conducir el aprendizaje del educando (fotos, láminas, videos, software, etc.).

Deben ser seleccionados adecuadamente, para que contribuyan a lograr un mejor aprendizaje y se deben tener en cuenta algunos criterios, por ejemplo:

- Deben ser pertinentes respecto de los objetivos que se pretenden lograr.
- Deben estar disponibles en el momento en que se los necesita.
- Deben ser adecuados a las características de los alumnos.
- Deben seleccionarse los recursos que permitan obtener los mejores resultados al más bajo costo, que impliquen la mínima pérdida de tiempo y puedan ser utilizados en distintas oportunidades.

El docente debe prever, seleccionar y organizar los recursos didáctico-pedagógicos que integrarán cada situación de aprendizaje, con la finalidad de crear las mejores condiciones para lograr los objetivos previstos. [6].

2.2.3 Software Educativo

En este apartado se explica el concepto, características, estructuras básicas, clasificación y funciones del software educativo, lo cual permitirá comprender con mayor facilidad las bases fundamentales del software educativo.

Los términos software educativo, programas educativos y programas didácticos son usados como sinónimos para designar genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje.

Esta definición engloba todos los programas que han estado elaborados con fin didáctico, desde los tradicionales programas basados en los modelos conductistas de la enseñanza, los programas de Enseñanza Asistida por Ordenador (EAO), hasta los aun programas experimentales de Enseñanza Inteligente Asistida por Ordenador (EIAO), que, utilizando técnicas propias del campo de los Sistemas Expertos y de la Inteligencia Artificial en general, pretenden imitar la labor tutorial personalizada que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los alumnos. [6].

2.2.3.1 Características Esenciales del Software Educativo

Los programas educativos pueden tratar las diferentes materias (matemáticas, idiomas, geografía, dibujo), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

- Son materiales elaborados con una finalidad didáctica, como se desprende de la definición.
- Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- Son interactivos.
- Individualizan el trabajo de los estudiantes.
- Son fáciles de usar. [6].

2.2.3.2 Estructuras Básicas del Software Educativo

La mayoría de los programas didácticos, igual que muchos de los programas informáticos nacidos sin finalidad educativa, tienen tres módulos principales claramente definidos: el módulo que gestiona la comunicación con el usuario (sistema input/output), el módulo que contiene debidamente organizados los contenidos informativos del programa (bases de datos) y el módulo que gestiona las actuaciones del ordenador y sus respuestas a las acciones de los usuarios (motor). [6]

El Entorno de Comunicación

Es el entorno a través del cual los programas establecen el diálogo con sus usuarios, y es la que posibilita la interactividad característica de estos materiales. Incluye:

- Las pantallas a través de las cuales los programas presentan información a los usuarios.
- Los informes y las fichas que proporcionen mediante las impresoras.
- El empleo de otros periféricos: altavoces, sintetizadores de voz, robots, módems, convertidores digitales-analógicos...
- El sistema de comunicación usuario-programa, que facilita la transmisión de información del usuario hacia el ordenador, incluye:
 - El uso del teclado y el ratón, mediante los cuales los usuarios introducen al ordenador un conjunto de órdenes o respuestas que los programas reconocen.
 - El empleo de otros periféricos: micrófonos, lectores de fichas, teclados conceptuales, pantallas táctiles, lápices ópticos, modems, lectores de tarjetas, convertidores analógico-digitales.

Las Bases de Datos

Las bases de datos contienen la información específica que cada programa presentará a los alumnos. Pueden estar constituidas por:

- Modelos de comportamiento. Representan la dinámica de unos sistemas.
- Datos de tipo texto, información alfanumérica.
- Datos gráficos. Las bases de datos pueden estar constituidas por dibujos, fotografías, secuencias de vídeo, etc
- Sonido. Como los programas que permiten componer música, escuchar determinadas composiciones musicales y visionar sus partituras.

El Motor o Algoritmo

El algoritmo del programa, en función de las acciones de los usuarios, gestiona las secuencias en que se presenta la información de las bases de datos y las actividades que pueden realizar los alumnos. Distinguimos los siguientes tipos de algoritmo:

- Lineal, cuando la secuencia de las actividades es única.
- Ramificado, cuando están predeterminadas posibles secuencias según las respuestas de los alumnos.
- Tipo entorno, cuando no hay secuencias predeterminadas para el acceso del usuario a la información principal y a las diferentes actividades. El estudiante elige qué ha de hacer y cuándo lo ha de hacer. Este entorno puede ser: Estático, Dinámico, Programable e Instrumental
- Tipo sistema experto, cuando el programa tiene un motor de inferencias y, mediante un diálogo bastante inteligente y libre con el alumno (sistemas dialogables), asesora al estudiante o tutoriza inteligentemente el aprendizaje.

2.2.3.3 Clasificación del Software Educativo

No obstante, de todas las clasificaciones la que posiblemente proporciona categorías más claras y útiles a los profesores es la que tiene en cuenta el grado de control del programa sobre la actividad de los alumnos y la estructura de su algoritmo, que es la que se presenta a continuación.[6]

Programas Tutoriales

Son programas que en mayor o menor medida dirigen, tutorizan, el trabajo de los alumnos. Pretenden que, a partir de unas informaciones y mediante la realización de ciertas actividades previstas de antemano, los estudiantes pongan en juego determinadas capacidades y aprendan o refuercen unos conocimientos y/o habilidades. Son programas que guían los aprendizajes de los estudiantes y facilitan la realización de prácticas más o menos rutinarias y su evaluación; A partir de la estructura de su algoritmo, se distinguen cuatro categorías: Programas lineales, programas ramificados, entornos tutoriales, sistemas tutoriales expertos.

Bases de Datos

Proporcionan unos datos organizados, en un entorno estático, según determinados criterios, y facilitan su exploración y consulta selectiva. Se pueden emplear en múltiples actividades como por ejemplo: seleccionar datos relevantes para resolver problemas, analizar y relacionar datos, extraer conclusiones, comprobar hipótesis.

Según la forma de acceder a la información se pueden distinguir dos tipos:

- **Bases de datos convencionales.** Tienen la información almacenada en ficheros, mapas o gráficos, que el usuario puede recorrer según su criterio para recopilar información.
- **Bases de datos tipo sistema experto.** Son bases de datos muy especializadas que recopilan toda la información existente de un tema concreto y además asesoran al usuario cuando accede buscando determinadas respuestas.

Simuladores

Presentan un modelo o entorno dinámico (generalmente a través de gráficos o animaciones interactivas) y facilitan su exploración y modificación a los alumnos, que pueden realizar aprendizajes inductivos o deductivos mediante la observación y la manipulación de la estructura subyacente; de esta manera pueden descubrir los elementos del modelo, sus interrelaciones, y pueden tomar decisiones y adquirir experiencia directa delante de unas situaciones que frecuentemente resultarían difícilmente accesibles a la realidad (control de una central nuclear, contracción del tiempo, pilotaje de un avión). Se pueden diferenciar dos tipos de simulador:

- **Modelos físico-matemáticos:** Presentan de manera numérica o gráfica una realidad que tiene unas leyes representadas por un sistema de ecuaciones deterministas.
- **Entornos sociales:** Presentan una realidad regida por unas leyes no del todo deterministas. Se incluyen aquí los juegos de estrategia y de aventura, que exigen una estrategia cambiante a lo largo del tiempo.

Constructores

Son programas que tienen un entorno programable. Facilitan a los usuarios unos elementos simples con los cuales pueden construir elementos más complejos o entornos. Se pueden distinguir dos tipos de constructores: Constructores específicos y Lenguajes de programación.

Programas Herramienta

Son programas que proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos... A parte de los lenguajes de autor (que también se podrían incluir en el grupo de los programas constructores), los más utilizados son programas de uso general que provienen del mundo laboral y, por tanto, quedan fuera de la definición que se ha dado de software educativo. Los programas más utilizados de este grupo son:

- Procesadores de textos,
- Gestores de bases de datos,
- Hojas de cálculo, Editores gráficos,
- Programas de comunicaciones,
- Programas de experimentación asistida,
- Lenguajes y sistemas de autor.

2.2.3.4 Funciones del Software Educativo

Funciones que pueden realizar los programas:

- **Función informativa.** La mayoría de los programas a través de sus actividades presentan unos contenidos que proporcionan una información estructuradora de la realidad a los estudiantes.
- **Función instructiva.** Todos los programas educativos orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a facilitar el logro de unos objetivos educativos específicos.
- **Función motivadora.** Generalmente los estudiantes se sienten atraídos e interesados por todo el software educativo, ya que los programas suelen incluir elementos para captar la atención de los alumnos, mantener su interés y, cuando sea necesario, focalizarlo hacia los aspectos más importantes de las actividades.
- **Función evaluadora.** La interactividad propia de estos materiales, que les permite responder inmediatamente a las respuestas y acciones de los estudiantes, les hace especialmente adecuados para evaluar el trabajo que se va realizando con ellos
- **Función investigadora.** Los programas no directivos, especialmente las bases de datos, simuladores y programas constructores, ofrecen a los estudiantes interesantes entornos donde investigar: buscar determinadas informaciones, cambiar los valores de las variables de un sistema, etc.
- **Función expresiva.** Dado que los ordenadores son unas máquinas capaces de procesar los símbolos mediante los cuales las personas representamos nuestros conocimientos y nos comunicamos, sus posibilidades como instrumento expresivo son muy amplias.
- **Función metalingüística.** Mediante el uso de los sistemas operativos (MS/DOS, WINDOWS) y los lenguajes de programación (BASIC, LOGO...) los estudiantes pueden aprender los lenguajes propios de la informática.

- **Función lúdica.** Trabajar con los ordenadores realizando actividades educativas es una labor que a menudo tiene unas connotaciones lúdicas y festivas para los estudiantes.
- **Función innovadora.** Aunque no siempre sus planteamientos pedagógicos resulten innovadores, los programas educativos se pueden considerar materiales didácticos con esta función ya que utilizan una tecnología recientemente incorporada a los centros educativos y, en general, suelen permitir muy diversas formas de uso. [6].

2.2.4 Materiales Educativos Computarizados

Bajo este nombre (abreviado MEC) se agrupan diversos tipos de aplicaciones encaminados a apoyar el aprendizaje.

Una referencia bastante apropiada es "Ingeniería de Software Educativo" de Alvaro Galvis, de donde se ha tomado la clasificación que se presenta. Una primera clasificación de herramientas y materiales para asistir el aprendizaje los divide en algorítmicos y heurísticos.

- **Materiales Algorítmicos:** predomina el aprendizaje vía transmisión de conocimiento desde quien sabe hacia quien lo desea aprender; quien diseña la herramienta planea secuencias de actividades para conducir al estudiante; el rol de alumno es asimilar el máximo de lo que se le transmite. Ej: Sistemas tutoriales, Sistemas de ejercitación y práctica.
- **Materiales Heurísticos:** predomina el aprendizaje por experimentación y descubrimiento; el diseñador crea ambientes ricos en situaciones que el alumno debe explorar; el alumno debe llegar al conocimiento a partir de la experiencia, creando sus propios modelos de pensamiento, sus propias

interpretaciones del mundo, las cuales puede someter a prueba con la herramienta. Ej: (Simuladores y Juegos educativos, Micromundos exploratorios, Sistemas expertos) [7].

2.2.5 Ingeniería de software educativo (ISE)

En la metodología de ISE desarrollada por Alvaro Galvis, los micromundos interactivos juegan un papel clave. Es a través de ellos como se crean ambientes lúdicos para aprender y es en ellos donde se viven experiencias que sirven de base para que el aprendiz genere o apropie conocimiento, dependiendo de la manera (algorítmica o heurística) como se use el micromundo. [8].

El micromundo interactivo es un ambiente digital representado a través de los procesos creativos e imaginativos del ser humano. La base de esta propuesta es crear micromundos altamente interactivos que tomen en cuenta el potencial tecnológico, pedagógico, psicológico y comunicativo de las disciplinas que se integran interdisciplinariamente para su construcción. [9]

2.2.5.1 Metodología para el Desarrollo de Software Educativo

La Figura 2.1, ilustra el flujo de acción en la metodología de ISE (1994) sobre la que se desea hacer incorporación del enfoque Orientado a Objetos. Como se aprecia, el ciclo de vida de una aplicación educativa puede tener dos maneras de ejecución, en función de los resultados de la etapa de análisis: en el sentido de las manecillas del reloj se procede a diseñar, desarrollar y probar lo que se requiere para atender una necesidad. En el sentido contrario, se someta a prueba aquello que se encontró puede satisfacer la necesidad. [8].

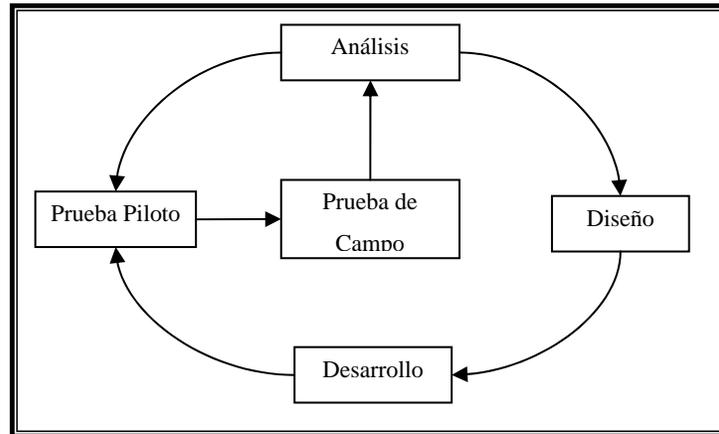


Fig. 2.1 Metodología ISE propuesta por Galvis.[Autor: Cadenillas R.]

2.2.5.2 La Ingeniería de Software y el Paradigma Orientado a Objetos (O.O.)

El enfoque de orientación por objetos (O.O.) es un paradigma que también cubre el ciclo de vida del software y que permite tener un mayor acercamiento al mundo que se modela y cómo funciona este mundo. En algunas metodologías de Ingeniería de Software (IS) se habla del análisis, diseño y desarrollo como tres procesos independientes cuya mezcla tiene como resultado final una aplicación que satisface X o Y necesidades.

El paradigma O.O. puede ser aplicado indistintamente al análisis, diseño o desarrollo de una aplicación. No es estrictamente necesario usar el enfoque en todas las etapas del ciclo de vida de una aplicación. Si se desea, se puede elaborar un buen análisis y diseño O.O., aún cuando la implementación no necesariamente siga el mismo esquema. Sin embargo, es una excelente alternativa usar O.O. en todo el ciclo de vida, buscando aprovechar al máximo todas las bondades de este nuevo paradigma. [8].

Características del Enfoque Orientado a Objetos (O.O.)

Con O.O. se puede hacer representación del mundo que se desea modelar en términos de los objetos que posee. Cada uno de ellos tiene sus propias características que lo identifican y un comportamiento específico. Estos aspectos pueden formalizarse con este enfoque. Con base en las características y comportamiento del objeto se pueden definir invariantes que deben cumplirse, permitiendo así verificar que el objeto funciona como se quiere.

Durante la definición de objetos del mundo se pueden usar los mecanismos de herencia y polimorfismo, para aprovechar las características y comportamiento de algunos objetos básicos, extendiéndolos para conseguir objetos con un comportamiento más específico. Además se puede usar otra importante característica llamada "reutilización de código", definiendo objetos que pueden ser usados en futuros desarrollos.

Ventajas del Enfoque Orientado a Objetos (O.O.)

Las ventajas de usar el enfoque O.O. se traducen en mejoramientos de calidad a lo largo del ciclo de vida de una aplicación, facilitando además el mantenimiento y la creación de nuevas versiones que extiendan el programa.

Al disminuir las barreras entre las etapas de análisis, diseño y desarrollo, se garantiza que se está hablando de las mismas cosas y en los mismos términos desde el comienzo del análisis hasta el final de la etapa de implementación. Esto evita inconsistencias y permite verificar que las cosas están claramente definidas y cumplen con todos los requerimientos, incluso antes de escribir una línea de código del programa. Las características anteriormente mencionadas (encapsulamiento,

herencia, reutilización) permiten crear un software mucho más robusto. Por último, el hecho de modelar el mundo y no únicamente los datos necesarios para determinada aplicación, permiten crear diversas aplicaciones sobre la misma información sin repetir los procesos de análisis de los mismos. Esto ofrece la posibilidad de dedicarse a cumplir con los requerimientos de la aplicación basándose en las facilidades que ofrecen los objetos del mundo ya modelados.

Se pueden enunciar varios beneficios de la aproximación orientada por objetos: reutilización de software: permite describir clases y objetos que podrán ser usados en otras aplicaciones; estabilidad: el diseñador piensa en términos de comportamiento de objetos, no en detalles de bajo nivel; diseño rápido y de alta calidad, puesto que se concentra en satisfacer los requerimientos y no en detalles técnicos; integridad; facilidad de programación al usar efectivamente toda la información de la fase de diseño, poniéndola en términos de un lenguaje específico; facilidad de mantenimiento, dado que al tener el modelo del mundo, es fácil realizar mantenimiento en términos de objetos, atributos y métodos de los mismos; independencia en el diseño, el diseño de un software se puede hacer independientemente de plataformas, software y hardware. [8]

2.2.5.3 Metodología ISE O.O.

Esta metodología busca unir todo lo anteriormente expuesto - metodología ISE con paradigma O.O - con miras a crear ambientes basados en micromundos interactivos. El gran reto es diseñar e implementar micromundos altamente interactivos que tomen muy en cuenta el potencial tecnológico y los recursos disponibles actualmente, sobre una sólida base educativa y comunicacional. Siguiendo el ciclo de vida de un MEC, la siguiente descripción permite entender cada una de sus etapas, enriquecidas con el enfoque OO mencionado. [8]

Análisis

El objetivo de esta etapa es determinar el contexto en el cual se va a crear la aplicación y derivar de allí los requerimientos que deberá atender la solución interactiva, como complemento a otras soluciones basadas en uso de otros medios (personales, impresos, audio-visuales, experienciales), teniendo claro el rol de cada uno de los medios educativos seleccionados y la viabilidad de usarlos. De acuerdo con Galvis en esta etapa se establece como mínimo la siguiente información:

- **Características de la población objetivo:** edad (física y mental), sexo, características físicas, y mentales (si son relevantes), experiencias previas, expectativas, actitudes, aptitudes, intereses o motivadores por aprender.
- **Conducta de entrada y campo vital:** nivel escolar, desarrollo mental, físico o psicológico, entorno familiar y escolar, etc.
- **Problema o necesidad a atender.** Para establecer la necesidad se puede recurrir a los mecanismos de análisis de necesidades educativas. Estos mecanismos usan entrevistas, análisis de resultados académicos, etc. para detectar los problemas o posibles necesidades que deben ser atendidas. Una vez identificado el problema se deben establecer las bases para resolverlo.
- **Principios pedagógicos y didácticos aplicables.** En esta fase se debe analizar cómo se ha llevado a cabo el proceso de enseñanza-aprendizaje para establecer cómo debe enfocarse el ambiente, qué factores tomar en cuenta, qué objetivos debe cumplir.
- **Justificación de uso de los medios interactivos como alternativa de solución.** Para cada problema o necesidad encontrada se debe establecer una estrategia de solución contemplando diferentes posibilidades. El apoyo informático debe ser tomado en cuenta siempre y cuando no exista un mecanismo mejor para resolver el problema: soluciones administrativas, ver

si el problema se soluciona al tomar decisiones de tipo administrativo; soluciones académicas, cambios en metodologías de clase; mejoras a los medios y materiales de enseñanza contemplando el uso de medios informáticos. [8]

Especificación de Requerimientos

Como síntesis de la etapa de análisis se deben formular los requerimientos que deberá atender el material interactivo que se desea obtener. La especificación de requerimientos debe contener lo siguiente:

- **Descripción de la Aplicación:** Contiene las características particulares de la aplicación dentro de determinado dominio: área de contenido, restricciones etc. Se hace una descripción de lo que hará la aplicación. Además se deben dejar claras las restricciones que tendrá y una descripción de los posibles escenarios de interacción que tendrá el usuario.
- **Diagramas de Interacción:** Permiten ver secuencias de interacción entre el usuario y la aplicación, representando lo que se espera del diálogo y dando más detalle a la descripción textual de la descripción de la aplicación. Los diagramas de interacción son un formalismo que permite ver la secuencia de acciones entre diferentes partes de la aplicación involucrada en llevar a cabo determinada actividad. [8]

Diseño

El diseño del Micromundo Interactivo se realiza a tres niveles diferentes: educativo, comunicacional y computacional. Al diseñar el ambiente en el que se desarrollará la acción se deben definir claramente los elementos que se determinaron como

necesarios en todo micromundo interactivo y aquellos deseables que convenga para el caso.

A continuación se define cada una de las etapas del diseño: diseño educativo, diseño comunicacional, diseño computacional. [8]

Diseño Educativo

Se debe establecer lo que hay que enseñar o reforzar para subsanar con apoyo del MEC las necesidades encontradas.

De acuerdo con Galvis, el diseño educativo debe resolver los siguientes interrogantes: ¿Qué aprender con el MEC? ¿En qué micromundo aprenderlo? ¿Cómo motivar y mantener motivados a los usuarios? ¿Cómo saber que el aprendizaje se está logrando?

¿Qué aprender con el MEC?

Para resolver este interrogante se debe partir del qué que subyace a micromundo: contenidos a tratar, derivados de las necesidades o problemas, tratando de detallar las unidades de contenido que van a tomarse en cuenta en el MEC. Se debe definir la red semántica que relaciona los conceptos que interesa desarrollar en la aplicación. Las relaciones de dependencia entre los diferentes temas deben tomarse en cuenta para no forzar el paso de un tema a otro y mantener coherencia a lo largo del material.

¿En qué ambiente o micromundo aprenderlo?

Para cada micromundo se debe establecer: Argumento, Mundo, Escenarios, Retos, Personajes y Herramientas, Objetos. Siguiendo el modelo O.O., se deben definir las clases que identifican cada uno de estos elementos. Al realizar el modelaje del mundo se deben definir las relaciones existentes entre estas clases.

¿Cómo motivar y mantener motivados a los usuarios?

En esta etapa del proceso de diseño se definen las metáforas usadas, así como cada personaje que aparece, dejando claro cuál es el rol que el usuario juega., las herramientas de interacción que podrá usar y cuál es el reto que debe resolver. En el caso de los micromundos interactivos es vital despertar motivación intrínseca proponiendo ambientes o situaciones que sean interesantes, que despierten curiosidad, que inviten al usuario a indagar a través de la experimentación con el micromundo. El micromundo debe ser novedoso y buscar sorprender al usuario, darle nuevas oportunidades de acción y plantear nuevos retos. Complementariamente se deben plantear retos que mantengan alerta a usuario en busca de pistas para resolverlos y con un nivel de complejidad apropiado.

¿Cómo saber que el aprendizaje se está logrando?

Las situaciones de evaluación (retos, etc.) deben estar relacionadas con los contenidos. La relevancia y pertinencia de determinado reto o prueba se debe sustentar con base en los contenidos que se han presentado y con la manera como han sido tratados.

- **Situaciones de evaluación:** El sistema de evaluación está relacionado con todos los retos del mundo. De acuerdo con esto debe definirse el nivel de logro para cada reto, que unido con todas las características (nivel de dificultad, tipo de aprendizaje, etc.) debe permitir evaluar qué ha hecho el usuario en el mundo y si lo hizo correctamente o no.
- **Manejo de retroinformación, refuerzo y niveles de logro:** Dependen mucho del enfoque del micromundo, según sea para aprendizaje por descubrimiento (enfoque heurístico) o por transmisión (enfoque algorítmico). Para decidir si el usuario ha logrado determinado nivel de aprendizaje se deben establecer criterios claros. Para esto deben extenderse todos los elementos del modelo computacional del mundo para que reaccionen ante determinados eventos.

Diseño Comunicacional

En esta fase del proceso de diseño se define la interfaz (zona de comunicación usuario-programa) de la aplicación. En este momento se debe complementar ese bosquejo definiendo formalmente los objetos que posee cada pantalla y cuáles elementos del mundo son usados/afectados. Es importante conseguir que la interfaz sea: amigable, flexible y agradable de usar; también debe ser consistente, es decir, cuidando que los mensajes y la distribución en pantalla, el juego de colores, etc. sigan un mismo patrón, también es necesario que sea altamente interactiva, lo cual conlleva tener mecanismos de comunicación entre el usuario y la aplicación.

Al definir la interfaz se debe tener en cuenta: ¿cuáles dispositivos de entrada-salida conviene poner a disposición del usuario para trabajar con el Micromundo?, ¿qué zonas de comunicación entre usuario y programa debe

tener el Micromundo?, ¿cuáles son las características de dichas zonas de comunicación?, ¿cómo verificar que la interfaz satisface los requisitos mínimos deseados? Para cada pantalla de la interfaz se deben definir las zonas de comunicación así como la distribución de las mismas.

Al diseñar una interfaz también se deben tomar en cuenta restricciones tecnológicas, características de la población y aspectos psicológicos de la percepción.

El modelo computacional de la interfaz consta de:

- Definición formal de cada pantalla.
- Objetivo.
- Eventos del modelo del mundo que está en capacidad de detectar.
- Diagrama de la pantalla, indicando cuáles objetos tiene y dónde están ubicados.
- Listado de las características tanto de la pantalla como de cada objeto (colores, tamaño de fuentes, resolución de imágenes, etc.).
- Enlaces con otros elementos de la interfaz. Notas adicionales. En caso de que se requiera realizar operaciones especiales en la interfaz..
- Diagrama de flujo de información en la interfaz. El diagrama indica la relación entre las diferentes pantalla de la interfaz. Con este diagrama se puede establecer cual es la secuencia que se seguirá en la aplicación.[8]

Diseño Computacional

Al final de esta etapa se tiene como resultado, claramente definidas, cada una de las diferentes clases de objetos, incluyendo sus atributos (indicando si

serán públicos -visibles a todo el mundo- o privados), el conjunto de métodos y el invariante de cada clase que corresponde al conjunto de restricciones o de requisitos que debe siempre cumplir una determinada clase.

Durante las fases de diseño educativo y comunicacional se han definido los diferentes objetos tanto del mundo como de la interfaz. Esta información se refina en esta fase, adecuándola a las posibilidades de la herramienta de desarrollo que se vaya a utilizar. Algunas clases necesitarán extenderse para ser usadas en el modelo.

Además se puede dar el caso de agregar nuevas clases y relaciones al modelo para dar mayor funcionalidad al modelo acorde con los requerimientos propios de la aplicación. La herramienta de desarrollo puede ofrecer mecanismos que faciliten la implementación de la interfaz. En caso de no ser así, el modelo del mundo se extiende de tal manera que pueda comunicarse efectivamente con el modelo de interfaz que deberá ser desarrollado.

Junto al conjunto de clases, llamado también modelo estático del mundo, se debe ilustrar la lógica acerca de cómo se desarrollan cada una de las actividades en el modelo. Para ello se deben refinar los casos de uso (algunos de los cuales ya se han obtenido en fases anteriores), ilustrando para cada uno de ellos el proceso que se sigue.

Esta información puede ayudar a redefinir el modelo antes de iniciar la fase de desarrollo. Además permite validar si el modelo es completo y permite satisfacer todos los requerimientos de la aplicación.

Desarrollo

En esta fase se implementa la aplicación usando toda la información obtenida anteriormente. Se toma la definición de clases y se implementa en el lenguaje escogido tomando en cuenta las restricciones computacionales que se tengan. Hay que establecer la herramienta de desarrollo sobre la cual se va a implementar la aplicación. Los criterios para escogerla incluyen; costo, disponibilidad en el mercado, portabilidad de la aplicación desarrollada, facilidades al desarrollador (ambientes gráficos de desarrollo, mecanismos de depuración, manejo de versiones, etc.).[8]

Prueba a lo largo y al final del desarrollo

La metodología propuesta permite ir depurando los componentes del modelo generado, haciendo validación con expertos de los prototipos durante la etapa de diseño y prueba uno a uno de los módulos desarrollados, a medida que estos están funcionales. Superada la depuración y ajuste, se pone a disposición una versión beta del micromundo interactivo. Esto conviene hacerlo con una muestra de la población; se pretende a través de dicha prueba piloto verificar que efectivamente la aplicación satisface las necesidades y cumple con la funcionalidad requerida. [8]

2.2.6 Multimedia

Los materiales multimedia son materiales informáticos interactivos que integran diversos elementos textuales (secuenciales e hipertextuales) y audiovisuales (gráficos, sonido, vídeo, animaciones...).

Según el tipo de interacción que ofrecen a los usuarios, estos programas o materiales educativos los podemos clasificar en dos grupos:

- **Documentos multimedia**, en los que la interacción se reduce a la consulta de los hipertextos y a un sistema de navegación que facilita el acceso a los contenidos.
- **Materiales multimedia interactivos**, que además facilitan otras interacciones con los usuarios (preguntas, ejercicios, simulaciones...).

Otra clasificación muy importante a tener en cuenta al considerar todos estos materiales es la que distingue los materiales propiamente didácticos de los que no lo son. Así podemos diferenciar:

- **Materiales multimedia de interés educativo**, materiales creados con una finalidad no educativa pero que en determinadas circunstancias pueden utilizarse igualmente como recursos didáctico con los estudiantes. Por ejemplo la página Web del Servicio Nacional de Meteorología o el CD del "Anuario El País".
- **Materiales multimedia didácticos**, que son los que han sido creados con la intención de facilitar determinados aprendizajes. Por ejemplo el programa del "cuerpo humano" (Z Multimedia) o la mayor parte de programas que realizan los profesores (premios PNTIC-CNICE). Además, según su estructura, estos materiales didácticos se pueden clasificar en programas tutoriales, de ejercitación, simuladores, bases de datos, constructores..., presentando diversas concepciones sobre el aprendizaje y permitiendo en algunos casos (programas abiertos, lenguajes de autor) la modificación de sus contenidos y la creación de nuevas actividades de aprendizaje por parte de los profesores y los estudiantes.[10]

2.2.7 Proceso Unificado De Desarrollo De Software

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo el proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par. [11]

2.2.7.1 Aspectos distintivos del Proceso Unificado

Están capturados en tres conceptos clave: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

- **El Proceso Unificado es dirigido por casos de uso.** Un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema. Sin embargo, los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de desarrollo.

- **El Proceso Unificado está centrado en la arquitectura.** El papel del arquitecto de sistemas es similar en naturaleza al papel que el arquitecto desempeña en la construcción de edificios. Similarmente, la arquitectura en un sistema de software es descrita como diferentes vistas del sistema que está siendo construido. La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. Ya que lo importante depende en parte del criterio, el cual a su vez viene con la experiencia, el valor de la arquitectura depende del personal asignado a esta tarea. Sin embargo, el proceso ayuda al arquitecto a enfocarse en las metas correctas, tales como claridad (understandability), flexibilidad en los cambios futuros (resilience) y reuso.
- **El Proceso Unificado es Iterativo e Incremental.** Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, esto es, deben ser seleccionadas y llevadas a cabo de una manera planeada. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus metas – y usualmente lo hace – el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.[11]

2.2.7.2 Fases del Proceso Unificado de Desarrollo de Software

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo se compone de varias fases, y dentro de cada una de ellas, los directores o los desarrolladores pueden descomponer adicionalmente el trabajo en iteraciones, con sus incrementos resultantes.

Las iteraciones de cada fase se desarrollan a través de las actividades de identificación de requisitos, análisis, diseño, implementación, pruebas e integración. Estas actividades son conocidas como flujos de trabajos fundamentales, y se describen a continuación:

- **Requisitos:** Flujo de trabajo fundamental cuyo propósito esencial es orientar el desarrollo hacia el sistema correcto. Esto se lleva a cabo mediante la descripción de los requisitos del sistema de tal forma que se pueda llegar a un acuerdo entre el cliente (incluyendo usuarios) y los desarrolladores del sistema, acerca de lo que la aplicación debe hacer y lo que no.
- **Análisis:** Flujo de trabajo fundamental cuyo propósito principal es analizar, refinar y estructurar los requisitos descritos en la actividad anterior (captura de requisitos).
- **Diseño:** Flujo de trabajo fundamental, en el cual, se formulan modelos que se centran en los requisitos no funcionales y el dominio de la solución. Además sirve de preparación para la implementación y pruebas del sistema.
- **Implementación:** Flujo de trabajo fundamental donde se implementa el sistema en términos de componentes, es decir, código fuente, guiones, archivos ejecutables, etc.
- **Pruebas:** Flujo de trabajo fundamental cuyo propósito fundamental es comprobar el resultado de la implementación mediante las pruebas de cada

construcción, incluyendo, tanto construcciones internas como intermedias, así como, las versiones finales del sistema. [11]

Fase de Inicio

Suele ser la fase más corta del desarrollo, y no debería alargarse demasiado en el tiempo. En caso contrario, podríamos encontrarnos en una situación de excesiva especificación inicial, yendo en contra del enfoque relativamente ágil del Proceso Unificado.

En esta fase se realizan las siguientes tareas:

- Desarrollar una descripción del producto final y presentar el análisis de negocio.
- Realizar una identificación inicial de riesgos.
- Establecen las principales funciones del sistema para los usuarios más importantes, la arquitectura a grandes rasgos y un plan de proyecto. [12]

Fase de Elaboración

Durante esta fase deberían capturarse la mayoría de requisitos del sistema, aunque los objetivos principales son tratar los riesgos ya identificados y establecer y validar la base de la arquitectura del sistema. Esta base se llevará a cabo a través de varias iteraciones, y servirá de punto de partida para la fase de construcción. La fase de elaboración termina, por tanto, al alcanzar el hito de la arquitectura del sistema. [12]

Fase de Construcción

Es la fase más larga del proyecto, y completa la implementación del sistema tomando como base la arquitectura obtenida durante la fase de elaboración. A partir de ella, las distintas funcionalidades son incluidas en distintas iteraciones, al final de cada una de las cuales se obtendrá una nueva versión ejecutable del producto. Por tanto, esta fase concluye con el hito de obtención de una funcionalidad completa, que capacite al producto para funcionar en un entorno de producción. [12]

Fase de Transición

En la fase final del proyecto se lleva a cabo el despliegue del producto en el entorno de los usuarios, lo que incluye la formación de éstos.

En lo relativo a la evolución del propio producto software:

- Gracias a las opiniones obtenidas de versiones preliminares, evoluciona desde la fase beta a una versión final.
- Se resuelven incidencias en la implantación e integración, y si existen, se clasifican aquellas que podrían justificar una nueva versión del producto.[12]

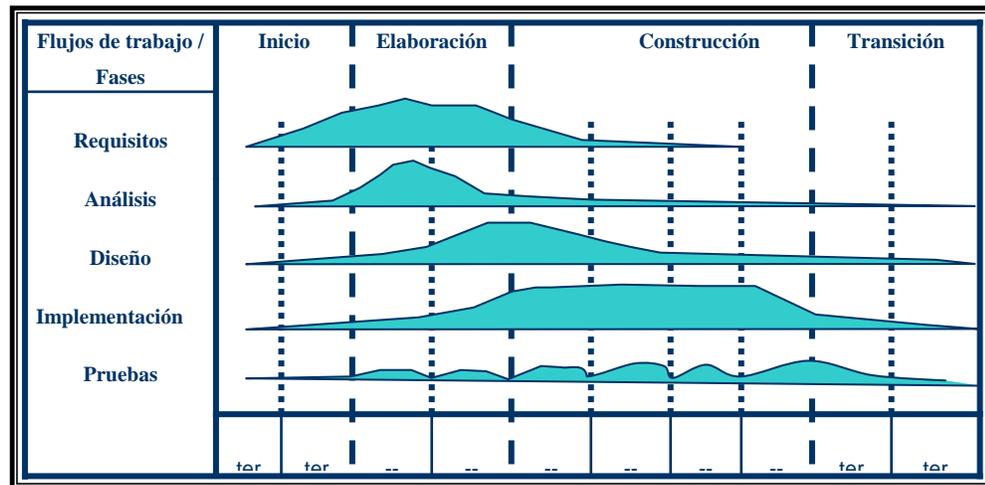


Fig. 2.2 Diagrama ilustrando como el énfasis relativo en las distintas disciplinas cambia a lo largo del proyecto.

La Figura 2.2 muestra las fases del proceso unificado de desarrollo de software y los flujos de trabajo que se desarrollan en cada una de las fases.

2.2.8 Lenguaje Unificado De Modelado (UML)

El Lenguaje Unificado de Modelado (UML – Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. [13]

2.2.8.1 Diagramas UML

A continuación se especifican cada uno de los diagramas UML.

Diagrama de Casos de Uso

Un diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relacionan con su entorno (usuarios u otras aplicaciones), es decir, representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

- **Elementos**

Actor: Es un usuario del sistema que necesita o usa algunos de los Casos de Uso; éste se comunica con el sistema mediante el envío y recepción de mensajes. Cabe destacar, que no todos los actores representan a personas; pueden ser actores otros sistemas o hardware. Un usuario físico puede actuar como uno o varios actores, desempeñando los papeles de esos actores en su interacción con el sistema. Se representa mediante una Figura humana dibujada con palotes, acompañado de un nombre significativo.

Casos de Uso: Es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de Casos de Uso representa la totalidad de operaciones desarrolladas por el sistema. Se representa en el diagrama por una elipse.

Relaciones de Uso: Entre los elementos de un diagrama de Casos de Uso se pueden presentar tres (3) tipos de relaciones:

1. **Comunica (communicates):** Es la relación entre un Actor y un Caso de Uso que denota la participación del actor en el Caso de Uso determinado.
2. **Usa (uses):** Es la relación entre dos Casos de Uso que denota la inclusión del comportamiento de un escenario en otro.
3. **Extiende (extends):** Es la relación entre dos Casos de Uso que denota cuando un Caso de Uso es una especialización de otro. [14]

Diagrama de Clase de Análisis

Es utilizado por los desarrolladores de software para determinar los requerimientos funcionales, considerando una o varias clases, o subsistemas del sistema a desarrollar.

Los Casos de Uso se describen mediante clases de análisis y sus objetos. El diagrama de Clase de Análisis se construye examinando los casos de usuarios, cerrando sus reacciones e identificando los roles de los clasificadores; éste está formado básicamente por los siguientes elementos: Clase de Interfaz, Clase de Control, Clase de Entidad y el Enlace. [15]

Diagrama de Colaboración

Este diagrama es utilizado para modelar interacciones entre objetos. En el análisis de este diagrama es más importante el paso de información de un objeto a otro, constituido por los mensajes que en su diseño se detallan.

Cada Caso de Uso se desarrolla como una realización de Casos de Uso, donde cada uno tiene un conjunto de clasificadores participantes que desempeñan diferentes roles. Cada Clase de Análisis representa un objeto o instancia de una clase en el diagrama de Colaboración donde se establece la cooperación existente entre ellas. La

secuencia en que estos objetos aparecen en un Caso de Uso se muestra usando números y comienza con un evento que llega a una interfaz, se sigue con un análisis de los eventos siguientes y la posible respuesta del sistema. [15]

Diagrama de Clase de Diseño

Se emplea para modelar la estructura estática de las clases en el sistema, sus tipos, sus contenidos y las relaciones que se establecen entre ellos. A través de este diagrama se definen las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización. [15]

- **Elementos**

Clase: Está representada por un rectángulo que dispone de tres (3) apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos. Cada clase debe tener un nombre único, que la diferencie de las otras.

Atributo: Representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase; éstos pueden representarse mostrando sólo su nombre, y su tipo, e incluso su valor por defecto. Pueden ser de tres (3) tipos:

1. **public (+):** Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
2. **private (-):** Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).

3. **protected (#):** Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven (herencia).

Método u Operación: Es la implementación de un servicio de la clase, que muestra un comportamiento común a todos los objetos. En resumen, es una función que le indica a las instancias de la clase que hagan algo. Pueden ser de tres (3) tipos:

1. **public (+):** Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
2. **private (-):** Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).
3. **protected (#):** Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven (herencia).

Relaciones entre Clases: Consiste en la interrelación que puede existir entre dos (2) o más clases con características y objetos diferentes. Estas relaciones son representadas por líneas entre las clases que presentan la cardinalidad; la cual indica el grado y nivel de dependencia entre las clases relacionadas; se escriben en cada extremo de la línea y pueden ser de tres (3) tipos: uno es a muchos (1...* ó 1...n), 0 es a muchos (0...* ó 0...n) o número fijo (m). Las relaciones entre las clases pueden ser:

- Relación de Asociación (): Asociación simple entre dos (2) clases. En ésta el tiempo de vida de un objeto no depende del otro.

- Relación de Agregación (): El objeto base utiliza al incluido para su funcionamiento, pero el tiempo de vida del objeto incluido es independiente del que lo incluye.
- Relación de Composición (): El objeto base se construye a partir del objeto incluido; el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye.[16]

Diagrama de Secuencia

El Diagrama de Secuencia del UML muestra la forma en que los objetos del sistema se comunican entre sí al transcurrir el tiempo. Éste consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

La idea primordial en este campo de visión expandido, es que las interacciones entre los objetos se realizan en una secuencia establecida y que la secuencia se toma su tiempo en ir del principio al fin.

- **Elementos**

Objetos: Representados por rectángulos colocados en la parte superior del diagrama, indican un conjunto coherente de roles que los usuarios desempeñan cuando interaccionan con los Casos de Uso.

Línea de Vida: Representada por una línea discontinua vertical que parte de cada uno de los objetos, indica la existencia de ese objeto a lo largo de un período de tiempo.

Activación: Representado por un rectángulo vertical en la línea de vida del objeto, indica el periodo de tiempo durante el cual un objeto está llevando a cabo una acción.

Mensajes: Representados por flechas horizontales entre los objetos; éstos permiten la comunicación entre ellos. [16]

Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue. [17]

Diagrama de Implementación

Un diagrama de implementación muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue): los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema, mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática. [18]

Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).
[19]

2.2.8.2 UML para la Web

La Ingeniería Web basada en UML, es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

A diferencia de los sistemas de software convencionales, las aplicaciones Web tienen características especiales, que van desde el entorno en el que operan hasta los

requerimientos de usuario. Para satisfacer estos aspectos especiales de las aplicaciones Web, UWE define vistas especiales representadas gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación. Además, UWE permite que un diseñador Web pueda también hacer uso de otra técnica de modelado UML que agreguen otras vistas de la aplicación, en otras palabras, UWE no limita el número de vistas posibles de una aplicación.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos. Con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. [23]

Actividades de Modelado

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de tareas y la visualización de escenarios Web para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. Estos modelos son los necesarios para lograr representar de manera satisfactoria los elementos arquitectónicamente significativos de una aplicación Web.[23]

2.2.8.3 WAE: Web Application Extensión

WAE extiende UML con estereotipos y restricciones para permitir el modelado de elementos de la arquitectura específicos de la web como parte del modelo del sistema. Se propone un amplio conjunto de estereotipos para la representación de elementos como páginas web, formularios, marcos, enlaces, servlets, etc., unas reglas de correcta formación de los modelos, y unas guías para el uso de la extensión. Está más

allá del propósito de este artículo explicar en detalle WAE, pero se darán algunos ejemplos de su aplicación.

En la tabla 2.1.siguiente se muestran los estereotipos y las relaciones para las clases, basados en WAE.

Tabla 2.1. Estereotipos y Relaciones WAE (1/2).

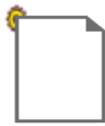
Estereotipos para las Clases	
Estereotipo	Descripción
 Server Page	Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor
 Client Page	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, scripts del lado del cliente, islas de datos, etc.
 Form	Representa una colección de campos de entrada que forman parte con una página del lado cliente (Client Page). Tiene una correspondencia directa con la etiqueta <FORM> de XHTML.
 ClientScript Object	Es una colección de scripts del lado del cliente que existe como un archivo separado y que son incluidos mediante una petición independiente por parte del navegador.

Tabla 2.1. Estereotipos y Relaciones WAE (2/2).

Estereotipos para las Relaciones entre las Clases	
Link	Representa un apuntador desde una "client page" hacia una "client page" o "server page". Corresponde directamente con una etiqueta <a> (ancla) de HTML
Submit	Esta relación siempre se da entre una "form" y una "server page", por supuesto, la "server page" procesa los datos que la "form" le envía (submits)
Build	Sirve para identificar cuales "server page" son responsables de de la creación de una "client page". Una "server page" puede crear varias "client page", pero una "client page" sólo puede ser creada por una sola "server page". Esta relación siempre es unidireccional
Redirect	Esta es también una relación unidireccional que indica que una página Web redirige hacia otra. En caso de que la página origen sea una "client page" esta asociación corresponderá con la "META" etiqueta y valor HTTP-EQUIV de "Refresh"*.

La Figura N° 2.3 muestra algunos de los estereotipos de WAE. Una página Cliente es una unidad de código HTML que es distribuida por el servidor web a sus clientes, que son los navegadores, usando el protocolo HTTP; los navegadores interpretan el código y presentan la información que contiene al usuario. Para obtener una Página Cliente, el navegador envía al servidor web, usando también HTTP, la dirección (URL, Uniform Resource Locator) de la página.

Una Página Servidora, por su parte, es una secuencia de comandos (que pueden ser CGI, ASP, JSP, etc.) que son procesados por el mismo servidor. Al igual que las

Páginas Cliente tienen una URL que es enviada por el navegador al servidor web, pero éste, en lugar de distribuir la página, ejecuta las instrucciones que contiene. Estas instrucciones pueden ser, por ejemplo, para consultar una base de datos y extraer de ella información que interesa al usuario del navegador, y terminan generalmente con la construcción de una Página Cliente que contiene la información obtenida, y que es enviada entonces por el servidor web al navegador para que se la presente al usuario. Desde el punto de vista de éste, el servidor web le envía la Página Cliente construida en forma dinámica, en respuesta a la URL de la Página Servidora enviada previamente.

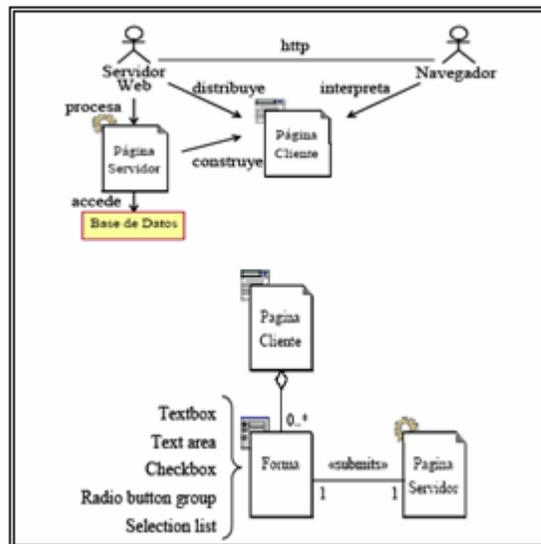


Fig 2.3 Estereotipos WAE.

Las Páginas Cliente pueden contener formularios, que son campos de entrada a través de los cuales el usuario ingresa información que luego es procesada en el servidor Web; dichos campos pueden ser cajitas de texto, áreas de texto con mayor capacidad, cajas de chequeo, grupos de botones de selección, y listas de selección. Cada formulario de la Página Cliente tiene asociada una Página Servidora que es la que recibe y procesa la información contenida en el formulario.

El Diagrama de Clases de Diseño de la Figura N° 2.4 utiliza WAE para representar el refinamiento del Diagrama de Clases de Análisis donde se muestran tanto las representaciones gráficas (íconos) de los estereotipos de las clases como los estereotipos de las relaciones entre éstas. La Página Cliente Acceso, que es la que se le presenta al Analista cuando activa la aplicación, posee el formulario ID_Clave que contiene las cajitas de texto para ingresar la identificación y la clave. Cuando se presiona el botón Aceptar, se transmite esta información al servidor web, que la entrega a la Página Servidora CtrlAcceso. El código contenido en esta página consulta Usuarios para verificar si es correcta la información ingresada por el Analista, y en tal caso construye la Página Cliente Menú, probablemente a partir de una plantilla a la que se agrega alguna información personalizada para el usuario. La página Menú ofrece al Analista opciones que lo llevan a otras páginas a través de enlaces de hipertexto. [24]

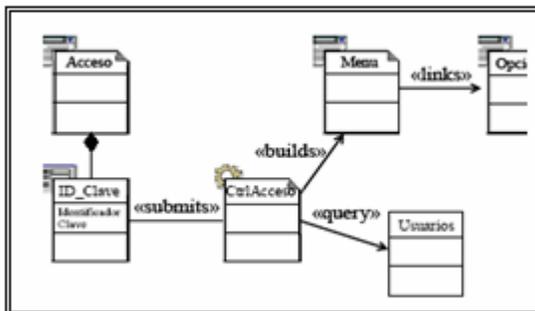


Fig 2.4 Diagrama de Clases de Diseño

2.2.9 Herramientas De Desarrollo

A continuación se presentan las herramientas que serán utilizadas para el desarrollo del Software.

2.2.9.1 Herramienta para creación de animaciones: FLASH

Es una tecnología para crear animaciones gráficas vectoriales independientes del navegador y que necesitan poco ancho de banda para mostrarse en los sitios Web. La animación en Flash se ve exactamente igual en todos los navegadores, un navegador sólo necesitan un plug-in para mostrar animaciones en Flash.

Con Flash los usuarios pueden dibujar sus propias animaciones o importar otras imágenes vectoriales. Macromedia Flash MX es la herramienta estándar de edición profesional para la creación de publicaciones Web de gran impacto. Tanto si crea logotipos animados, controles de navegación de sitios Web, animaciones de gran formato, o aplicaciones Web o sitios Web completos usando Flash, descubrirá que por su capacidad y flexibilidad. [25]

2.2.9.2 Lenguaje de creación de páginas Web: HTML

HTML es una implementación del standard SGML (Standard Generalized Markup Language), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones. Metalenguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones.

Características

- Permite crear lenguajes de codificación descriptivos.
- Define una estructura de documentos jerárquica, con elementos y componentes interconectados.
- Proporciona una especificación formal completa del documento.

- No tiene un conjunto implícito de convenciones de señalización. Soporta, por tanto, un conjunto flexible de juegos de etiquetas.
- Los documentos generados por él son legibles.[26]

2.2.9.3 Lenguaje de Acceso a Base de Datos: SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet. SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para crear y manipular directamente las bases de datos Jet y crear consultas SQL de paso a través para manipular bases de datos remotas cliente - servidor.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

SQL nos permite trabajar con cualquier tipo de lenguaje (ASP o PHP) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL).
[27]

2.2.9.4 Lenguaje de Programación para Páginas Web: PHP

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones.

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. [28]

2.2.10 Contenido Programático de la Asignatura Modelos de Operaciones I

Unidad I: Análisis de redes PERT-CPM

1. Aspectos generales: Importancia del análisis de redes.
2. Fases de planeación de una red PERT-CPM.
3. definición de actividades y relaciones de precedencia.
4. Representación y elaboración de la red: Actividades ficticias.
5. Análisis de la red: Método de revisión hacia delante y método de revisión hacia atrás. Ruta crítica.
6. Red PERT-CPM con incertidumbre: Variabilidad en los tiempos de las actividades. Variabilidad en la fecha del proyecto. Probabilidad asociada a una fecha específica de terminación de un proyecto.
7. Pert-Costos.

Unidad II: Teorías de Juegos.

1. Aspectos básicos

2. Formulación de juegos de dos personas con suma cero: Características del juego. Matriz de pago o de recompensa.
3. Solución de juegos con estrategias mixtas.

Unidad III: Programación Dinámica.

1. Definición. Características de la programación dinámica. Principios de optimización de Bellman.
2. Naturaleza recursiva de los cálculos en programación dinámica.
3. Elementos básicos de un modelo de programación dinámica: definición de las etapas. Definición de las alternativas en cada etapa. Definición de los estados para cada etapa.
4. Recursión en reversa y en avance.
5. Modelo de la ruta más corta.
6. Modelo del tamaño de la fuerza de trabajo.
7. Aplicaciones de la programación dinámica.

Unidad IV: Programación no lineal.

1. Generalidades de la teoría de optimización clásica: Introducción a la optimización.
2. Presentación de un problema de programación no lineal. Funciones convexas y cóncavas.
3. Optimización no restringida: Condiciones necesarias y suficientes para extremos.
4. Optimización restringida.
5. Método de multiplicadores de Lagrange: Restricciones de igualdad. Algoritmos para restricciones no negativas. Algoritmos para restricciones de desigualdad.

CAPÍTULO III: FASE DE INICIO

3.1 INTRODUCCIÓN

De acuerdo a lo descrito en el Proceso Unificado de Desarrollo de Software, en la fase de inicio se pone en marcha el proyecto, mediante, de los flujos de trabajo de requisitos, análisis, diseño, implementación y, en un menor grado, pruebas, sin embargo, en este capítulo, se hace mayor énfasis en la captura y el análisis de los requisitos fundamentales del Software Educativo de Modelos de Operaciones I (SEMOI).

Durante la fase de inicio se identifican los requerimientos, tanto, funcionales como, no funcionales del software, a través del estudio de las necesidades educativas de la población objetivo (población a la que está dirigido el software). Además, se realiza el análisis de la viabilidad del proyecto, estableciendo y priorizando los riesgos. De acuerdo a estos requisitos se diagraman los casos de uso más importantes, así como, sus actores y escenarios.

Para el flujo de trabajo de análisis, en esta fase, se realizan las actividades de análisis de la arquitectura y análisis de los casos de uso, permitiendo realizar un esbozo del diseño de la arquitectura candidata.

Tomando como base los requisitos y los riesgos más críticos analizados, en esta fase, se planifica el proyecto y se crea un plan detallado de trabajo para la fase siguiente.

A esta etapa corresponde el análisis educativo, en el cual, se recopilan los requisitos y características de la población objetivo.

3.2 FLUJO DE TRABAJO: REQUISITOS

En la etapa de requisitos se obtiene una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre qué debe y qué no debe hacer el sistema. De esta manera es consigue el objetivo fundamental de esta fase: la obtención del modelo de casos de uso. Para ello, se debe seguir una serie de pasos que permita la más efectiva captura de los requisitos.

En este flujo de trabajo se incluye la enumeración de los requisitos candidatos, el establecimiento del contexto del sistema, captura de los requisitos funcionales y, finalmente, captura de requisitos no funcionales, como los pasos fundamentales utilizados como punto de partida para el desarrollo de la aplicación.

3.2.1 Identificación de Requisitos Candidatos

La enumeración de los requisitos candidatos nos permite implementar una versión futura del sistema. Se establece una lista de características que deben tomarse en cuenta para el desarrollo del software, basándonos en la experiencia de docentes y profesores, en el sistema educativo tradicional, en la demanda de materiales educativos computarizados y en las características de la población objetivo. A continuación se describe los requisitos candidatos del SEMOI:

- Identificar el conjunto de conocimientos previos necesarios para la comprensión de la asignatura.

- Organizar el contenido de la asignatura Modelos de Operaciones I, siguiendo la estructura (por unidades) del programa establecido por el Departamento de Computación y Sistemas de la Universidad de Oriente.
- Dividir cada capítulo en lecciones que puedan ser aplicadas y comprendidas mejor por los estudiantes.
- Adaptar la información, de manera, que pueda ser presentada mediante el uso de técnicas multimedia y la World Wide Web.
- Desarrollar actividades interactivas y animaciones que den apoyo al proceso de enseñanza de cada uno de los capítulos.
- Establecer mecanismos de evaluación, que permitan apreciar el nivel de conocimiento adquirido por el alumno después de cada lección.
- Determinar mecanismos que permitan, al usuario, el rápido acceso a la información.
- Definir elementos multimedia que ofrezcan características atractivas para el usuario de la aplicación.

Aunado a lo expuesto, se requiere la comprensión del contexto del sistema, para estimar las necesidades que debe satisfacer el software educativo.

3.2.2 Contexto del Sistema

Para capturar los requisitos de forma correcta y para realizar correctamente el diseño del sistema, es necesario conocer el contexto en el cual se ubica el sistema.

Para expresar, de manera utilizable, el contexto del sistema, se requiere de un modelo de dominio, donde se describan los conceptos importantes del contexto, como objetos, y se definan las relaciones entre estos objetos. A medida que analicemos y

diseñemos el sistema, los objetos de dominio, nos ayudaran a identificar algunas de las clases del sistema.

El modelo de dominio se describe en la Figura 3.1 mediante un diagrama de UML, específicamente el diagrama de clases de dominio. En este diagrama se representan todos los objetos del sistema mediante rectángulos, las relaciones entre estos objetos se identifican mediante líneas, a las cuales se le asocia un nombre y una cardinalidad, representada mediante: un asterisco (*) cuando son varios objetos, un uno (1) cuando se trata de un solo objeto y (1...*) si se trata de uno o varios objetos. La asignatura a la cual se hace referencia en el diagrama es Modelos de Operaciones I. En el Modelo de Dominio se muestran los tres tipos de asociaciones: Simple (relaciones entre dos objetos donde no existe una dependencia, se representa mediante una línea recta ) , agregada (relaciones que expresan un acoplamiento más fuerte entre las clases, un objeto requiere del otro, se representa a través de un rombo vacío ) y compuesta (constituye un acoplamiento muy fuerte entre las clases, existe una dependencia entre los objetos, se identifica con ) un rombo relleno).

Por ejemplo, el objeto alumno expresa una asociación simple con el objeto asignatura con una cardinalidad de varios a uno, es decir, varios alumnos estudian esa asignatura, al mismo tiempo se establece una asociación compuesta entre asignatura y unidad, con una cardinalidad de uno a varios, es decir, una asignatura posee varias unidades, por otra parte el rol “Posee”, entre unidad y lección indica una asociación agregada y en la cardinalidad se expresa que una unidad puede tener una o varias lecciones.

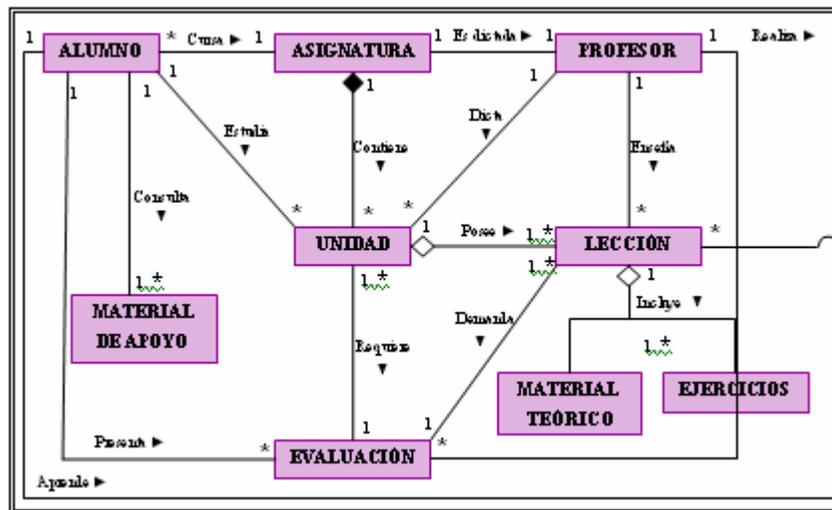


Fig. 3.1 Diagrama de Clases de Dominio

3.2.3 Requisitos Funcionales

Los requisitos funcionales definen las acciones fundamentales que va a realizar el sistema. Se determinan una vez que se logra la comprensión del contexto del sistema del SEMOI. Para el establecimiento de los requisitos funcionales es necesario realizar el análisis educativo a la población objetivo (profesores y alumnos) de la asignatura Modelos de Operaciones I. Estos requisitos funcionales se representan como casos de uso.

3.2.3.1 Análisis de Necesidades Educativas de La Población Objetivo

Para el análisis de las necesidades educativas de la población a la que va dirigido el software, se deben identificar los aspectos relevantes del sistema educativo actual de la asignatura Modelos de Operaciones I en la Universidad de Oriente, estableciendo sus debilidades y/o deficiencias, y tomando igualmente los aspectos positivos, con el fin de determinar posibles causas y soluciones, a las necesidades encontradas

mediante un estudio de las características de la población. Este estudio se basa en observaciones directa del entorno, entrevistas, aplicación de encuestas a alumnos y profesores, entre otras.

El análisis de necesidades educativas se lleva a cabo mediante las siguientes actividades, que se realizan de forma consecutiva: consultas a fuentes de información apropiadas, identificación del problema educativo, análisis de posibles causas de los problemas detectados, análisis de alternativas de solución, justificación del uso de medios interactivos, establecimiento del papel del computador y selección o planeamiento del MEC (material educativo computarizado)

Consulta a Fuentes de Información Apropriadas

Una apropiada fuente de información sobre necesidades educativas es aquella que está en capacidad de indicar fundamentadamente las debilidades o problemas que se presentan, o se pueden presentar, para el logro de los objetivos de aprendizaje en un ambiente de enseñanza-aprendizaje dado.

Los profesores y alumnos son fuentes de información primaria para detectar y priorizar aspectos problemáticos; ellos más que nadie saben en qué puntos el contenido, el modo o los medios de enseñanza, se están quedando cortos frente a las características de los estudiantes y a los requerimientos del currículo que guía la acción.

Para recolectar la información necesaria se analizan, mediante encuestas (Ver Anexo A), las opiniones de profesores de Modelos de operaciones I de la Universidad de Oriente y de los alumnos que ya han cursado dicha asignatura. La encuesta se aplico a la población objetivo, que incluyó a los 2 profesores de la materia y a 40

alumnos del Departamento de Computación y Sistemas que cursaron Modelos de operaciones I.

A continuación se presentan los resultados de las encuestas realizadas.

Encuestas Realizadas a Estudiantes que Cursaron la Asignatura

La encuesta permite conocer la opinión de los estudiantes con respecto a como fue dictada la asignatura y a la posibilidad de incluir un software educativo multimedia e interactivo para dar apoyo al proceso de enseñanza- aprendizaje de la asignatura. El sondeo generó los siguientes resultados:

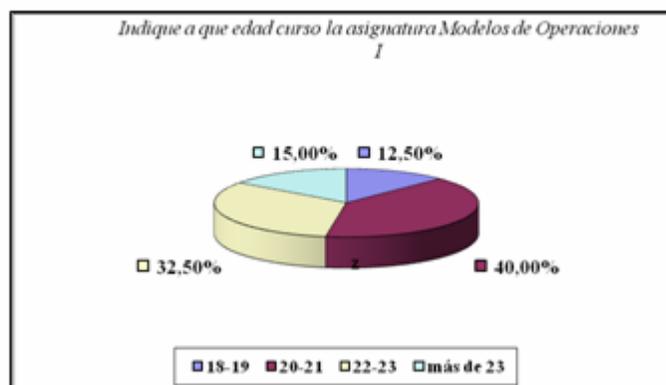


Fig. 3.2 Intervalos de edad de los estudiantes para cursar Modelos de Operaciones I

Según la Figura 3.2, la mayoría de los estudiantes cursa la materia entre los 20 y 21 años, equivalente al 40% de los entrevistados, existiendo, también, alta incidencia entre 22 y 23 años con un 32%.

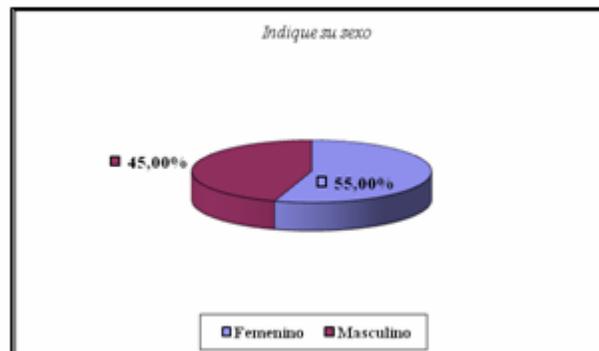


Fig. 3.3 Sexo de la población Objetivo

La mayoría de la población encuestada pertenece al sexo masculino con un 55%, mientras que el sexo femenino representa el 45% restante. Esto se observa en la Figura 3.3.

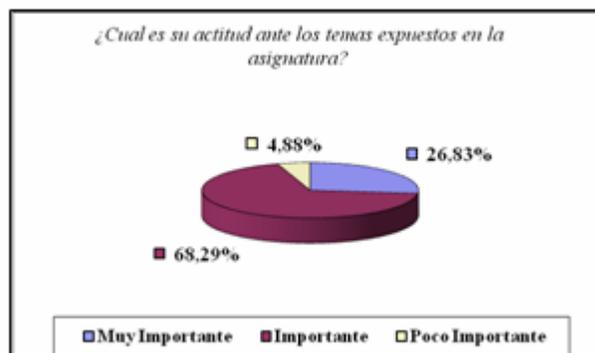


Fig. 3.4 Opinión de los Estudiantes acerca de la importancia de Modelos de Operaciones I.

Según los estudiantes encuestados, la materia Modelos de Operaciones I es considerada primordial para su formación como ingenieros. Siendo “Muy Importante”, para el 26,83 %, “Importante” para 68,29% de los entrevistados y “Poco importante” para el 4,88% restante. Como se muestra en la Figura 3.4.

La asignatura Modelos de Operaciones I, es interesante para la mayoría de los encuestados (ver Figura 3.5), ya que, se evidencia que el 85% del porcentaje total están “Interesados” con el contenido expuesto en la asignatura y el 12,50% están “Muy Interesados”, completando casi la totalidad de la población objetivo.

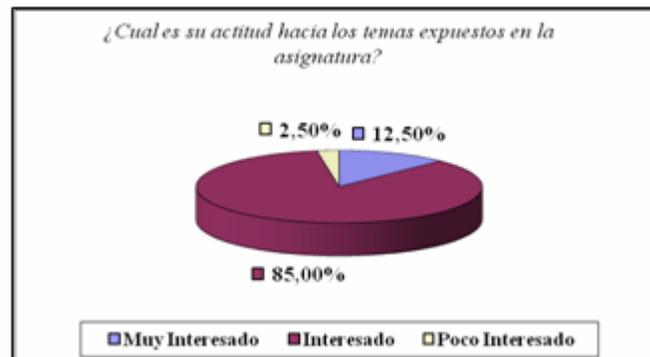


Fig. 3.5 Interés de los estudiantes en la asignatura.



Fig. 3.6 Porcentaje de alumnos que aprendieron o no, lo necesario en la materia.

La Figura 3.6 revela que un 77,50% de los estudiantes encuestados aprendieron lo necesario mediante el curso de la asignatura, sin embargo, existe un porcentaje considerable (22,50%) que no lo hizo.

Según los estudiantes, las explicaciones poco claras del profesor (40%) seguida de la no explicación de todo el contenido de la materia (27,5%) fueron las principales

situaciones que influyeron en su rendimiento al momento de cursar la asignatura. Ver Figura 3.7.

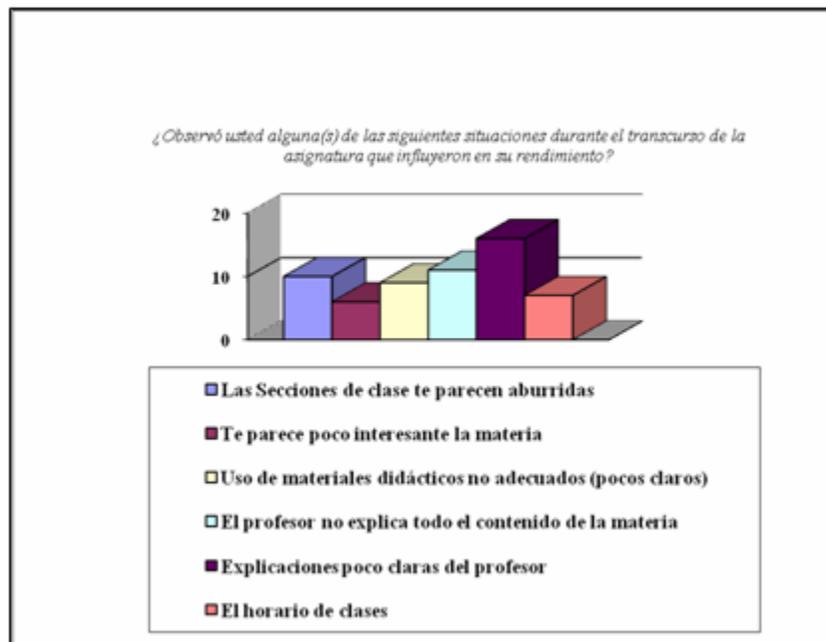


Fig. 3.7 Situaciones que influyen en el rendimiento de los alumnos

La Figura 3.8 demuestra que la principal herramienta utilizada por los estudiantes para el estudio de la asignatura fue el libro con 67,5%. Otra herramienta que también tomó importancia en sus estudios fueron los apuntes de clase con un 45%.

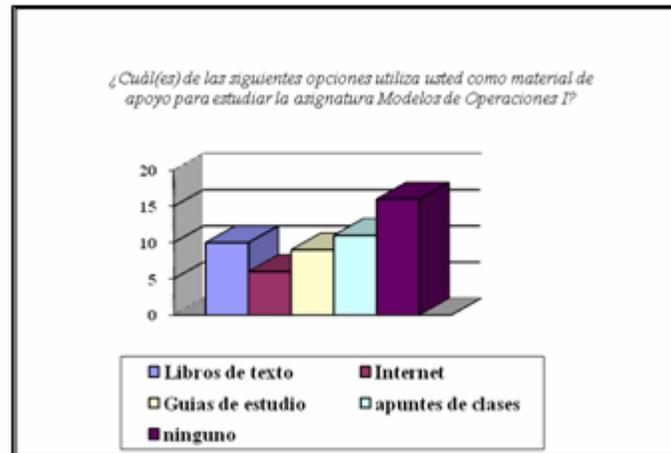


Fig. 3.8 Material de apoyo usado por los estudiantes

La mayoría de los estudiantes considera optimización de operaciones como una asignatura necesaria para cursar Modelos de operaciones I con 67,5%, seguida de Estadística I con 50%. Estos resultados se observan en la Figura 3.9.

Por otra parte, la Figura 3.10 indica que existe una paridad del 50 %, entre los alumnos que consideran suficientes los materiales de apoyo usados y los que consideran que no son suficientes

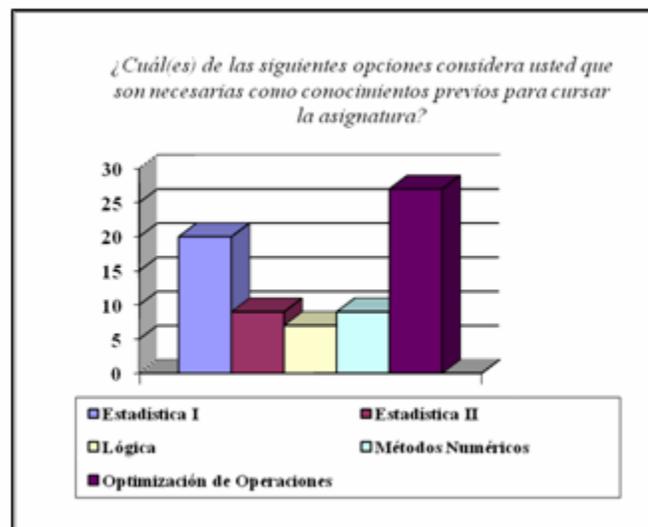


Fig. 3.9 Conocimientos previos para Modelos de Operaciones I

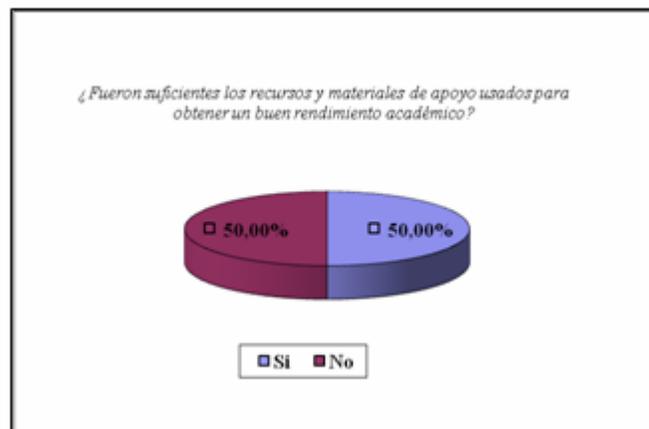


Fig. 3.10 Porcentaje de alumnos que consideran suficientes los materiales de apoyo usados

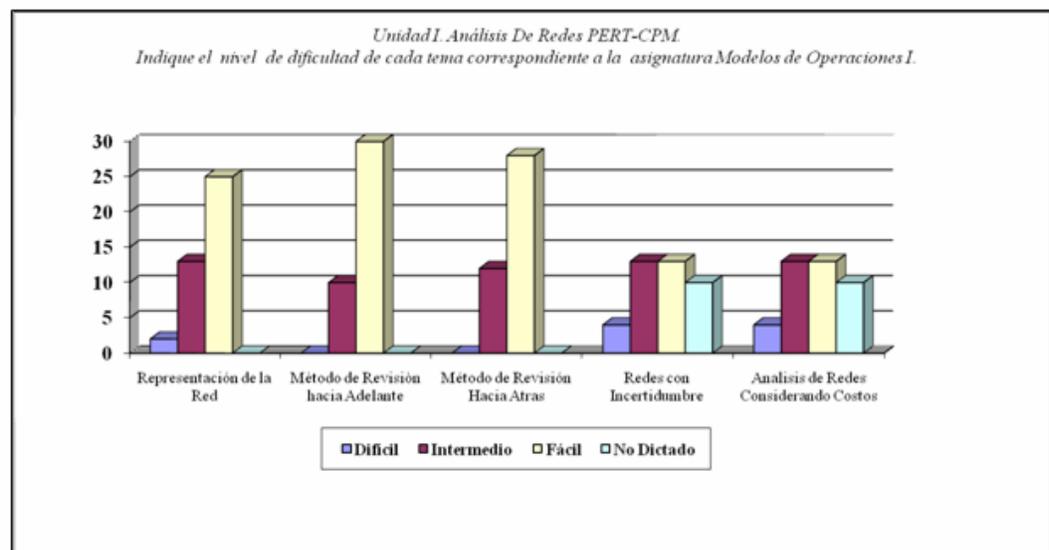


Fig. 3.11 Nivel de dificultad de la unidad I

Para analizar los niveles de dificultad en la unidad I de la asignatura se consulta la Figura 3.11, la cual, demuestra que es relativamente fácil para los encuestados. Ciertos estudiantes le asignan a los 2 últimos contenidos (análisis de redes con

incertidumbre y análisis de redes considerando costos) un nivel de dificultad entre intermedio y difícil.

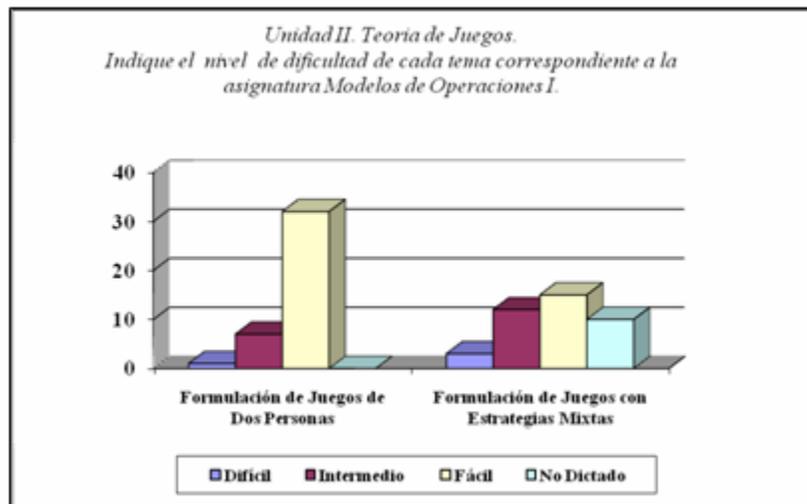


Fig. 3.12 Nivel de dificultad de unidad II

La Figura 3.12, señala que la unidad II es considerada fácil por la mayoría de los estudiantes, tanto en el contenido 1 como en el contenido 2. Mientras que, la Figura 3.13, expone claramente que el nivel de dificultad se incrementa en la Unidad III, ya que, en la mayoría de los contenidos el grado de dificultad es intermedio o alto (difícil).

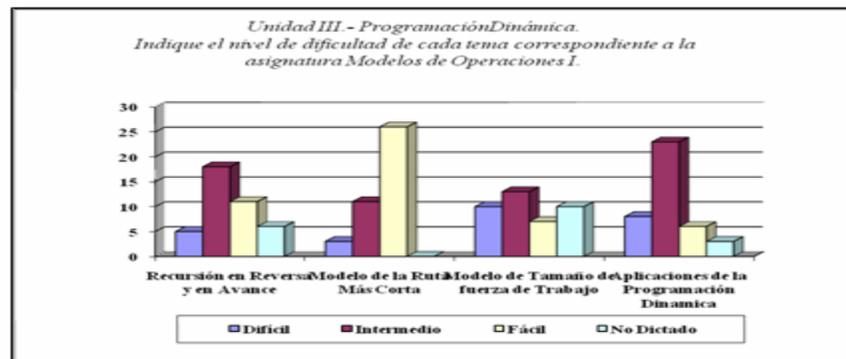


Fig. 3.13 Nivel de dificultad de unidad III

Para terminar con el análisis de las unidades, tenemos la Figura 3.14 donde la unidad IV se caracteriza por tener un grado de dificultad bajo o intermedio en todos los contenidos.

El 90% de los encuestados considera necesario la inclusión de un software educativo multimedia para dar apoyo al proceso enseñanza-aprendizaje de Modelos de Operaciones I, mientras que un 10% no estuvo de acuerdo con la creación del software. (Ver Fig.3.15)

Para el caso de las personas que consideran necesario el desarrollo de un software educativo para la asignatura, se analizan las Figuras 3.16 y 3.17 indicando lo siguiente:

Para la Figura 3.16 la población objetivo, el software debe contener todos los temas expuestos en la asignatura, siendo el “Análisis de redes PERT-CPM” el de mayor demanda con 67,5%, seguido por programación dinámica y programación no lineal (47,5%).

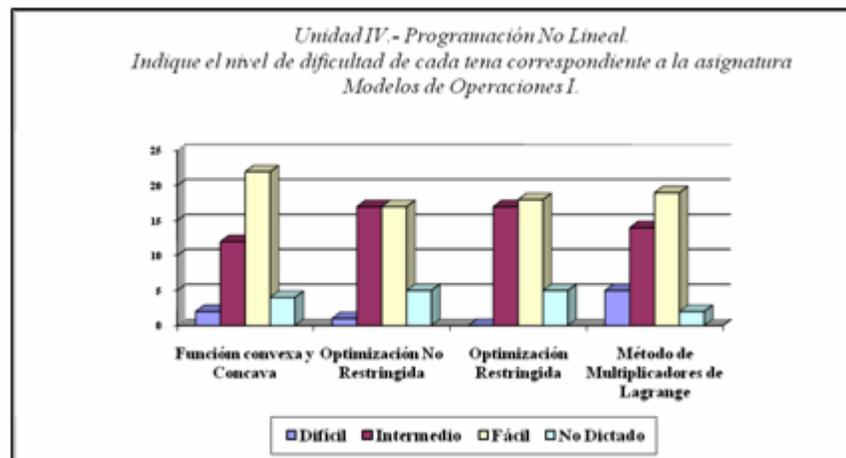


Fig. 3.14 Nivel de dificultad de unidad IV

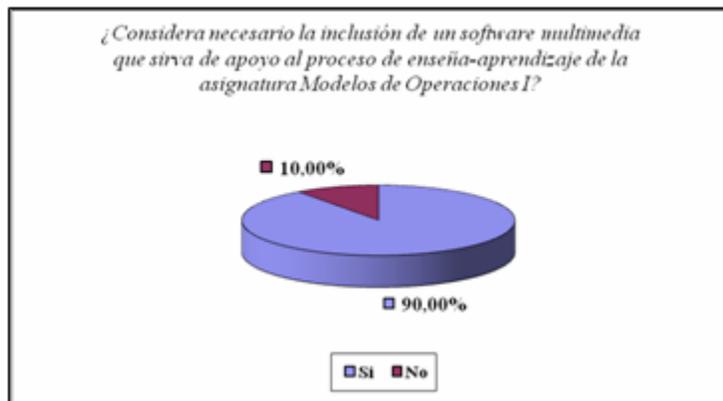


Fig. 3.15 Necesidad de un software educativo, según los estudiantes

Según el sondeo realizado a los estudiantes, el software debe contener imágenes (55%), sonidos (32,5%), animaciones (57,5%), video (37,5%), juegos interactivos (37,5%). Resultados obtenidos de la Figura 3.17.

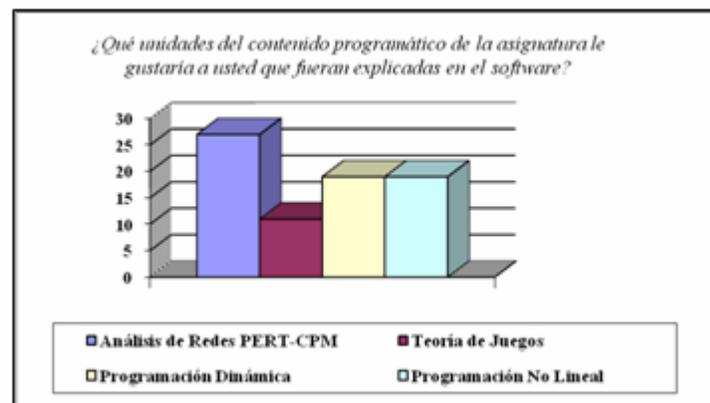


Fig. 3.16 Temas que deberían incluirse en el software multimedia

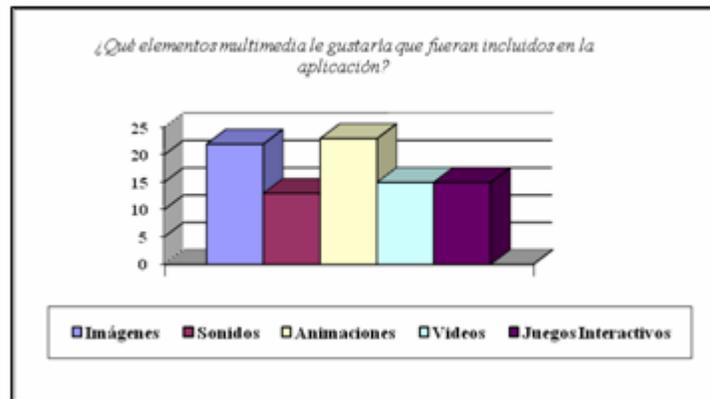


Fig. 3.17. Elementos multimedia preferidos por los estudiantes para el Software multimedia

Encuestas Realizadas a Profesores que Dictan la Asignatura

Esta encuesta permite conocer la opinión de los profesores con respecto al desenvolvimiento de los alumnos en clase y, además, permite conocer si consideran necesario la implementación de un software educativo para la asignatura que imparten.

Actualmente el Departamento de Computación y Sistemas cuenta con dos profesoras para dictar la materia Modelos de operaciones I. La encuesta fue aplicada a ambas profesoras, las cuales respondieron lo siguiente:

- Las dos profesoras están de acuerdo en que la materia Modelos de Operaciones I, es muy importante.
- El nivel de dificultad del contenido en cada unidad es considerado el siguiente:
 - Unidad I: Una de las profesoras considera que todo el contenido de la unidad I es fácil, la otra profesora piensa que el primer y quinto

contenido (Representación de la red y análisis de redes considerando costos, respectivamente) tienen un nivel de dificultad intermedio y un nivel de dificultad fácil para los contenidos restantes para esta unidad.

- Unidad II: Ambas profesoras están de acuerdo en que el contenido de esta unidad es fácil.
 - Unidad III: Al igual que en la unidad II, para esta unidad ambas profesoras también están de acuerdo pero para este caso se considera que todo el contenido de la unidad tiene nivel de dificultad intermedio.
 - Unidad IV: nuevamente las profesoras coinciden en sus respuestas y señalan que todo contenido de la unidad V, es relativamente fácil.
- Una de las profesoras utiliza como material de apoyo para dictar clases, las exposiciones realizadas por los alumnos, mientras que la otra profesora hace uso de la pizarra / marcador, guías para dictar las clases y un software para realizar practicas de la asignatura.
 - Al conocer el nivel de participación de los estudiantes en el transcurso de las clases, una profesora responde que la participación del estudiantado es de un nivel medio, mientras que la otra profesora considera que el nivel de participación es alto.
 - Con respecto al rendimiento de los alumnos en las diversas unidades de la asignatura, una de las profesoras indica que los alumnos presentan un rendimiento alto para las unidades I y II y un rendimiento medio para las unidades III y IV. Mientras que la otra profesora señala q los alumnos presentan un rendimiento medio para las unidades I, II y IV y un rendimiento bajo para la unidad III.
 - Una de las profesora considera que la desmotivación es una causa atribuida al bajo rendimiento de los alumnos para ciertos temas, mientras, que la segunda

profesora piensa que la dificultad para comprender el tema y falta de conocimientos previos son las causas atribuidas al bajo rendimiento de los alumnos.

- En una de las encuestas se revela que utilizar un material didáctico, hacer la clase más dinámica y realizar horarios de clases adecuados son opciones que se pueden realizar para mejorar el rendimiento del alumno, por lo contrario la segunda encuesta deja ver que solo es necesario utilizar un material didáctico.
- Ambas profesoras encuestadas consideran necesario un Material Educativo Computarizado (MEC) que facilite el estudio de los temas de mayor complicación de la asignatura Modelos de Operaciones I.
- Ambas profesoras consideran que los ejemplos y las autoevaluaciones son elementos que debería contener la aplicación para la mejor comprensión de la asignatura, pero una de las profesoras considera que además de los elementos ya mencionados también se debe agregar animaciones a la aplicación.
- Las dos profesoras encuestadas consideran que si puede ser cubierta la asignatura durante un semestre.

Análisis de las Encuestas

Luego de efectuar el estudio de las encuestas realizadas a profesores y alumnos, podemos concluir que:

- La edad promedio de los estudiantes para cursar la asignatura Modelos de Operaciones I, se ubica entre 20 y 21 años de edad. Existiendo una población predominante de hombres.
- La asignatura es considerada, por alumnos y profesores, de vital importancia para la formación de ingenieros en el Departamento de Computación y Sistemas.

- Los alumnos se muestran interesados con el contenido programático dictado en clases. Esto genera a su vez, que cada alumno aprenda lo necesario con mayor facilidad.
- Los alumnos y profesores coinciden en que no cuentan con el tiempo necesario para explicar todo el contenido de la materia debido, principalmente, a la constante interrupción de clases en la universidad.
- El libro es la herramienta más usada por los estudiantes, para el estudio de Modelos de Operaciones I.
- La mayoría de los estudiantes consideran que, para cursar modelos de Operaciones I, son fundamentales los conocimientos de optimización de operaciones y estadística, sin embargo, un porcentaje menor de estudiantes consideran que los conocimientos obtenidos en métodos numéricos y lógica son necesarios para la comprensión de la asignatura.
- Es evidente la necesidad de materiales de apoyo que complementen la educación impartida por el profesor en salón de clase.
- La asignatura Modelos de Operaciones I no resulta muy complicada para los estudiantes, sin embargo, existen ciertos temas que representan un grado alto de dificultad, los cuales, requieren de mayor dedicación y de materiales que apoyen y contribuyan con la mejor comprensión del contenido.
- La creación e implementación de un software educativo multimedia de Modelos de Operaciones I, es necesaria, según estudiantes y profesores, para reforzar el proceso de enseñanza que se lleva a cabo en la Universidad de Oriente. Según los encuestados, la aplicación debería incluir todos los temas de la asignatura, principalmente el análisis de redes PERT-CPM, ya que, esta es la unidad más demandada por los estudiantes, además de esto, su contenido se presta para el desarrollo de actividades interactivas.

- Un software multimedia debe incluir todos los elementos que permitan facilitar el aprendizaje, entre estos se incluyen imágenes, sonidos, videos, animaciones, juegos y evaluaciones interactivas.

Identificación del Problema Educativo

Para la identificación del problema educativo tomamos como base el análisis realizado a las encuestas. Como resultado de esta etapa se debe contar con una lista priorizada de problemas en los distintos temas u objetivos que componen un plan de estudio, con anotación de la fuente o evidencia de que existe cada problema y de la importancia que tiene resolverlo.

La consulta a fuentes de información apropiadas nos permite identificar los problemas que presenta el sistema educativo existente. Estos problemas se describen a continuación:

- **Falta de motivación en los alumnos.** Ciertos estudiantes se muestra interesados con la asignatura y la consideran importante en su carrera sin embargo, admiten que necesitan motivación, ya que, la clase puede tornarse aburrida, debido a la falta de elementos que incrementen su interés.
- **Constantes pérdidas de clases:** Debido a los frecuentes inconvenientes que se presentan en la Universidad de Oriente, la mayoría de las veces, se deben suspender clases, esta situación influye en la secuencia normal de la clase, provocando falta de interés, disminución de horas de clases, entre otras.
- **Falta de dedicación y atención.** Siendo una asignatura con un grado de dificultad intermedio, los estudiantes no le prestan la debida atención y dedicación, razón por la que pueden obtener un bajo rendimiento.

- **No se explica la totalidad del contenido de la asignatura.** Existe un tiempo limitado para impartir cada tema de la asignatura, que podría no ser suficiente para completar la totalidad de contenido requerido por el Departamento de Computación y Sistemas de la Universidad de Oriente.
- **Falta de material didáctico interactivo que facilite la comprensión de la asignatura.** Los estudiantes no cuentan con gran variedad de herramientas que den apoyo y refuercen la educación impartida por el profesor.

Análisis de Posibles Causas de los Problemas Detectados

Para poder atender las necesidades o resolver los problemas detectados, es imprescindible saber a qué se debieron y qué puede contribuir a su solución. En particular interesa resolver aquellos problemas que están relacionados con el aprendizaje, en los que eventualmente un MEC podría ser de utilidad.

Un problema de rendimiento, o de aprendizaje, puede deberse a muchas razones, en este caso podemos mencionar las siguientes:

- **Los alumnos no prestan la debida atención y no participan en clase.** Son frecuentes las distracciones de los alumnos. Aunado a esto, no participan en clase, ya sea por confusión en cuanto al contenido del tema, temor a equivocarse, entre otras.
- **Los materiales didácticos usados por alumnos y profesores son poco efectivos.** Los profesores no disponen de los recursos didácticos multimedia de calidad para impartir las clases (ejemplo: video beam). Los alumnos no cuenta con materiales lo suficientemente buenos para facilitar el aprendizaje. Son escasos los materiales bibliográficos disponibles.

- **El tiempo dedicado al estudio no es suficiente.** Ciertos alumnos estudian 1 o 2 días antes del examen, tiempo que no es suficiente (en la mayoría de los casos) para la buena comprensión del tema estudiado.
- **La metodología usada por el profesor no es muy atractiva para el estudiante.** Se hacen necesarias herramientas más atractivas para incrementar el interés y motivación de los alumnos en la asignatura.

Análisis de Alternativas de Solución

Dependiendo de sus causas, algunos problemas o necesidades, se pueden resolver tomando decisiones administrativas o académicas.

- **Administrativas:** Mejorar los ambientes de aprendizaje y asignar más tiempo al estudio de la asignatura. Si los estudiantes no traen los conocimientos de base, pueden tomarse medidas administrativas como son impedir que avancen en el currículo mientras no se nivelen.
- **Académicas:** Conseguir los medios y materiales que hagan posible disponer de los ambientes de aprendizaje apropiados, así como capacitar los profesores en el uso de estos nuevos medios, promover un mayor trabajo individual de los estudiantes sobre los materiales para aprendizaje cuando se trata de preparar nuevas ayudas educativas o de mejorar la calidad de las pruebas académicas, mejoras en los medios y materiales de enseñanza convencionales, como son los materiales impresos, guías de estudio, así como los materiales y las guías de trabajo o de laboratorio. También existirá la posibilidad de utilizar otros medios no tan convencionales. Una solución computarizada debe considerarse como complemento más que como sustituto de una práctica, una etapa del proceso de aprendizaje a partir del objeto de conocimiento.

Tomando como base lo antes expuesto, dentro de las soluciones académicas se determinó que el desarrollo de un software educativo de Modelos de Operaciones I, sería de gran apoyo en el proceso de enseñanza-aprendizaje de la asignatura, ya que, facilitará la labor del docente y despertará motivación e interés al estudiante, creando micromundos excitantes cuya exploración los conduzca a un amplio nivel de dominio del tema.

En cuanto a las soluciones administrativas, dependen directamente del estudiante y de la Universidad De Oriente, del presupuesto que ésta tenga asignado para tal fin y de la disposición de las autoridades correspondientes de proporcionar alternativas multimedia al alumnado de la institución.

Justificación del Uso de Medios Interactivos

El uso de medios interactivos es hoy un factor indispensable y protagónico a la hora de realizar el plan de educación multimedia. Las nuevas tecnologías acercan en tiempo real al estudiante al contenido de la asignatura. Adicionalmente, permite el aprovechamiento y la administración de esa información, por parte de los estudiantes, a ritmo y secuencia propia.

Interactuar con un sistema experto que es de interés o con un simulador puede ayudar a desarrollar criterio, a refinar el conocimiento, pero no sustituye la labor del docente.

Estas características pueden resultar muy provechosas para impartir clases de Modelos de Operaciones I en la Universidad de Oriente, ya que, es una materia práctica que se presta para actividades interactivas. Además, los medios interactivos generaran interés y motivación en el alumnado.

Con la implementación de un software educativo interactivo para la asignatura Modelos de Operaciones I, cada estudiante tiene la posibilidad de aprender por medio de una herramienta que le parezca atractiva, donde se expone el contenido de la asignatura en su totalidad por medio del uso de gráficos, sonidos, imágenes, animaciones, etc., y que facilita el aprendizaje permitiendo que se desenvuelva a su propio ritmo y que realice actividades evaluativas que definan si el aprendizaje se realiza de manera exitosa.

Establecimiento del Papel del Computador

El siguiente paso es determinar el papel del computador como una alternativa idónea para la resolución de las necesidades educativas analizadas.

Si las posibilidades que brindan las herramientas de propósito general (un procesador de texto, una hoja de cálculo, un graficador, un manejador de bases de datos) no son adecuadas o son insuficientes, habrá que pensar en qué otro tipo de ambiente educativo informático es conveniente. Tratándose de necesidades educativas relacionadas con el aprendizaje, según la naturaleza de éstas, se puede establecer el papel del computador para la aplicación a realizar y qué tipo de MEC conviene usar.

Una vez analizadas los requisitos se establece que el computador es una herramienta adecuada para facilitar el proceso enseñanza - aprendizaje de la asignatura Modelos de Operaciones I, mediante un software educativo. Además, nos permite desarrollar una aplicación en red, a través, de la World Wide Web.

Selección o Planeamiento del Mec (Material Educativo Computarizado)

Seguidamente, se debe establecer si es necesario el desarrollo de un MEC que satisfaga la necesidad que se ha detectado.

Luego de analizar las necesidades de la población objetivo, se determina que es necesario un Material Educativo Computarizado que facilite la comprensión del contenido de la asignatura Modelos de Operaciones I. Este material debe adaptarse a las exigencias de la Universidad de Oriente, en cuanto al contenido programático de la asignatura, y además, debe cubrir los requerimientos expresados, mediante encuestas y entrevistas, realizadas a alumnos y profesores.

Este MEC se traduce como un software educativo para facilitar el proceso enseñanza-aprendizaje de la asignatura Modelos de Operaciones en el Departamento de Computación y Sistemas de la Universidad de Oriente.

3.2.3.2 Especificación de los Requerimientos

Una vez, realizado el análisis de las necesidades educativas de la población objetivo, se requiere el establecimiento de una herramienta educativa que refuerce los conocimientos de los estudiantes y sirva de apoyo al profesor en el proceso de enseñanza de la asignatura Modelos de Operaciones I.

El software educativo multimedia debe ser interactivo y presentar micromundos atractivos e interesantes que mantengan al estudiante motivado. A su vez, debe ser de fácil uso, ofrecer explicaciones multimedia detalladas de todo el contenido de la asignatura, así como, juegos y actividades interactivas.

Se requiere que cada estudiante tenga la oportunidad de aprender a su propio ritmo, teniendo la posibilidad de adelantar o retroceder en el contenido según sea su deseo.

Para establecer si el aprendizaje se esta realizando de forma correcta, se requiere de autoevaluaciones que permitan al estudiante valorar la efectividad de

educación que esta obteniendo. Esta evaluación se basa en preguntas y respuestas almacenadas en una base de datos.

3.2.3.3 Identificación de Riesgos

Para el desarrollo del software educativo de Modelos de Operaciones I, es necesaria la identificación de los riesgos o probabilidades de que ocurran sucesos adversos al proyecto, que podrían afectar el correcto funcionamiento de la aplicación. Los riesgos críticos encontrados son los siguientes:

- **Accesibilidad.** La interfaz podría no ser lo suficientemente sencilla o clara para el usuario (no se entiende la interfaz), evitando que acceda fácilmente a la información. Esto trae como consecuencia, que no se aprovechen al máximo los beneficios del software. Para mitigar este riesgo se debe crear una interfaz sencilla y fácil de entender y explorar por el usuario, además se deben crear micromundos interactivos con opciones de ayuda que ofrezcan al usuario información sobre el ambiente donde se encuentre.
- **Robustez.** El software debe ser robusto para que su funcionamiento sea correcto si se presentan condiciones anormales. Por lo tanto, su diseño, estructura y funcionamiento deben brindar una clara perspectiva del sistema, ofreciendo flexibilidad al cambio, reutilización y comprensión.
- **Fallas en el sistema de almacenamiento.** El sistema podría presentar fallas con la información almacenada, generando datos erróneos al usuario. Esta situación afecta la veracidad de la información proporcionada al usuario, quien no es capaz de distinguir entre lo correcto e incorrecto. Para evitar esta situación se debe revisar el contenido presentado en la aplicación antes de ser mostrado al usuario.

- **Bajo rendimiento del hardware.** Se deben recomendar las características necesarias de hardware que permita una eficiente ejecución del software educativo, ya que este cuenta con un conjunto de imágenes, animaciones e información que pueden generar sobrecargas en la aplicación. El uso de un equipo que no cumpla los requisitos establecidos puede traer como consecuencia que el usuario se fastidie o se canse debido a que probablemente se verá afectado el tiempo de respuesta y la velocidad de la aplicación.
- **Sistema de motivación.** Se debe realizar un software con características interactivas suficientes, que mantengan al usuario interesado en el contenido de la asignatura. Si la aplicación carece de elementos interactivos que motiven al estudiante, este no prestará la atención necesaria al tema y el aprendizaje le resultará monótono. Este riesgo se controla incluyendo mecanismos multimedia e interactivos como imágenes, videos, juegos, animaciones y sonidos que estimulen el estudio.
- **Disponibilidad del servidor:** La aplicación se desarrolla bajo un entorno Web y para su aplicación mediante Internet se requiere de la completa disponibilidad del servidor asignado. Esto permitirá responder a las peticiones de los usuarios en un instante dado. Si el servidor falla o no se encuentra disponible no se podrá transferir correctamente la información al cliente.

3.2.3.4 Características del Proceso de Enseñanza - Aprendizaje

En el proceso enseñanza-aprendizaje de la asignatura de modelos de Operaciones I, se emplea la metodología tradicional para dictar la clase. Generalmente, el profesor prepara con anticipación el material a exponer y dicta su clase valiéndose del pizarrón y marcadores, para las dos horas prácticas y teóricas establecidas en los pensums actuales de Ingeniería de Sistemas e Ingeniería en Computación de la Universidad de Oriente, aprobados en el año 2004. Además, el profesor tiene la responsabilidad de

aclarar las dudas de los alumnos y estimular su participación en clase, también asigna horas de consultas donde los estudiantes pueden exponer las inquietudes que surgen durante sus horas de estudio.

Durante la clase, el alumno realiza anotaciones del material presentado por el profesor, también efectúa preguntas sobre conceptos no comprendidos y, si lo considera necesario, asiste a sus horas de consulta.

Las unidades son evaluadas por el profesor mediante diversas modalidades como trabajos de investigación, exposiciones, talleres y pruebas escritas. Estas evaluaciones se realizan de forma grupal o individual dependiendo de lo establecido por el profesor, del tema y del tiempo disponible para realizar todas las evaluaciones y concluir con el contenido de la asignatura.

3.2.3.5 Característica de la Población Objetivo.

El SEMOI se desarrolla para todas aquellas personas interesadas en la educación de Modelos de Operaciones I. A continuación, se establecen las características de la población, a la cual, está dirigido el software.

- Estudiantes del 7mo semestre de Ingeniería de Sistemas e Ingeniería en Computación de la Universidad de Oriente. Siendo Modelos de Operaciones I, una asignatura obligatoria para Ing. de Sistemas y electiva para Ing. en computación.
- Estudiantes de sexo femenino y masculino.
- Edades comprendidas entre los 19 y 24 años.
- Profesores de la Universidad de Oriente que dicten asignaturas relacionadas con la investigación de operaciones.

- Estudiantes y profesores de otras instituciones.
- Estudiantes con conocimientos previos de: Optimización de operaciones, estadística I, estadística II, lógica, métodos numéricos.

3.2.3.6 Representación de los Requisitos como Casos De Uso

Para completar los requisitos funcionales del sistema se debe establecer quienes serán los usuarios y que debe hacer el sistema para cumplir con sus necesidades. Esto se realiza definiendo los elementos principales del modelo de casos de uso, es decir, los actores y los casos de uso.

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso representa una unidad de interacción entre un usuario (humano o máquina) y el sistema. Cada Caso de Uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto. Un Caso de Uso puede 'incluir' la funcionalidad de otro Caso de Uso o puede 'extender' de otro Caso de Uso con su propio comportamiento. Los casos de uso típicamente se relacionan con 'actores'. Un actor es un humano o una máquina que interactúa con el sistema para realizar un trabajo significativo, mediante envío y recepción de mensajes.

Actores

Los actores presentes en el SEMOI son resultado del estudio de la los requisitos funcionales y de la información obtenida del contexto del sistema a través del modelo de dominio. Se establecen 3 actores para el sistema, los cuales se describen a continuación:

Tabla 3.1. Actores del Sistema (1/2).

ACTORES	DESCRIPCIÓN	
Usuarios	Estudiantes	Utilizan el software como apoyo usando las herramientas didácticas, mediante las actividades de aprendizaje, consultas y evaluaciones (por cada tema).
	Profesores	Consultan la aplicación como guía para impartir sus clases.
	Interesados	Todas aquellas personas que se encuentran en la red, buscando información acerca de los temas contenidos en el software.

Tabla 3.1. Actores del Sistema (2/2).

ACTORES	DESCRIPCIÓN
Administrador	Es el encargado de cargar y manejar los escenarios y eventos producidos por los usuarios y el sistema. Este actor representa a las herramientas de Macromedia, el lenguaje PHP y servidor WEB, que permite el desarrollo de la aplicación.
Base de datos	Representa la colección de datos que da soporte a las evaluaciones y a los usuarios del sistema.

Casos de Uso

Cada caso de uso identifica las acciones que realizan los actores del sistema. Para la fase de inicio se establecen los siguientes casos de uso:

Tabla 3.2. Casos de Uso (1/2)

NOMBRE	DESCRIPCIÓN
Navegar Contenido de Unidades	El usuario visualiza y selecciona una de las cuatro unidades que contiene el software, lo que permitirá activar un sub-menú que contiene las lecciones, actividad didáctica y evaluación de cada unidad.
Explorar Unidad	Permite al usuario interactuar con la lección seleccionada de la unidad activada, mostrando la interfaz que permite el acceso a cada lección, actividad didáctica y evaluación.
Realizar Actividad Didáctica	Actividades de aprendizaje disponible para cada unidad que permiten reforzar los conceptos básicos de cada lección.

Tabla 3.2. Casos de Uso (2/2)

NOMBRE	DESCRIPCIÓN
Evaluar Unidad	Este caso de uso muestra una serie de preguntas a responder que se identifican con números, al seleccionar una de las preguntas se despliegan varias respuestas, donde el usuario tendrá la oportunidad de elegir la correcta. Además contará con la opción de visualizar su puntuación, una vez respondidas todas las preguntas.
Estudiar Lección	Contiene Información de la asignatura Modelos de Operaciones I, con conceptos prácticos que permiten al usuario la fácil comprensión de la asignatura.
Observar Animación	Se muestran imágenes con movimiento que permiten exponer de una forma más sencilla el contenido de la asignatura.
Efectuar Juegos	Actividades didácticas educativas que, a través de ejemplos prácticos, exponen el contenido de la asignatura.
Obtener Ayuda	Indica los pasos necesarios para el manejo del software: las funciones

Del Sistema	de los botones, el sistema de navegación de la aplicación y demás acciones ejecutadas por el software.
Consultar Diccionario	Proporciona una breve definición de aquellos términos poco claros, que se contemplaran en el software.
Configurar Evaluación	El administrador realiza la configuración y actualización de las evaluaciones de las unidades del software. Las cuales son almacenadas en la base de datos.

Diagrama de Casos de Uso

En la Figura 3.18 se muestra el diagrama de casos de uso del SEMOI. Los óvalos representan los casos de uso y los muñecos simbolizan a los actores del sistema.

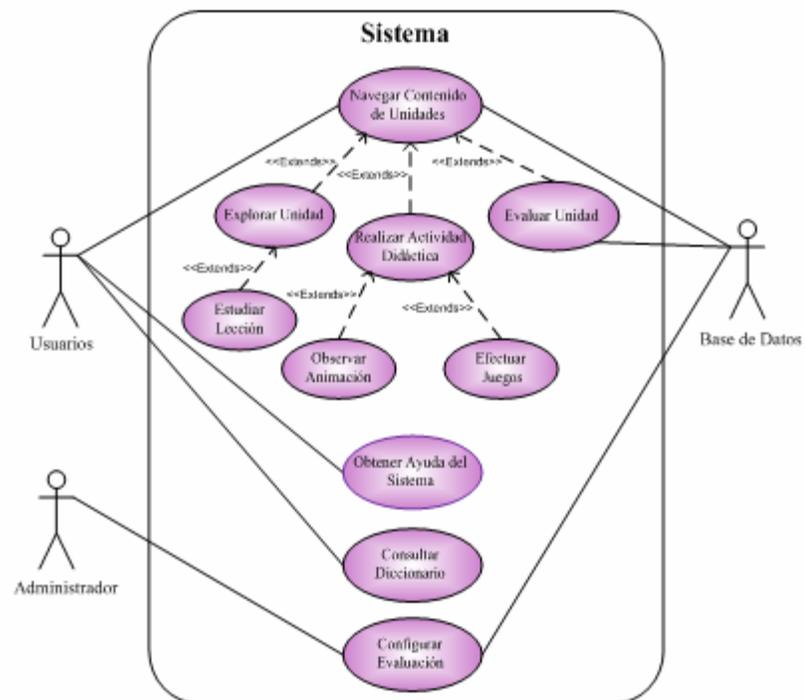


Fig. 3.18. Diagrama de Casos de Uso

En el diagrama, se observa que el actor “usuario”, Navega por el Contenido de las Unidades, Obtiene Ayuda del Sistema y Consulta el Diccionario si lo requiere. Al Navegar por el Contenido de las Unidades, el usuario, tiene acceso a los casos de uso Explorar Unidad, Evaluar Unidad y Realizar Actividad Didáctica, que extiende, a su vez, a los casos de uso Observar Animación y Efectuar Juegos. Igualmente Explorar Unidad extiende a caso de uso Estudiar Lección. El “administrador” es el encargado de Configurar las Evaluaciones del sistema. Los casos de uso Navegar Contenido de Unidades, Evaluar Unidad y Configurar Evaluaciones, cuando son activados por el usuario y administrador respectivamente, realizarán el proceso que permite la conexión con la “base de datos” que contiene las evaluaciones de cada unidad.

3.2.4 Requisitos No Funcionales

Los requisitos no funcionales deben establecer restricciones en el software que esta siendo desarrollado, en el proceso de desarrollo y en restricciones específicas que el software pueda tener. Describe no lo que el software hará, sino como lo hará. Los requisitos no funcionales son difíciles de verificar, y por ello son evaluados subjetivamente.

Estos requisitos especifican propiedades del sistema como restricciones del entorno o de la implementación, rendimiento, dependencia de la plataforma, facilidad de mantenimiento, extensibilidad, fiabilidad, entre otros. En este caso se consideran los siguientes requisitos no funcionales:

- **Entorno.** La presentación de la información se debe realizar usando diferentes herramientas como texto, sonido, imágenes y/o videos, con colores y movimientos llamativos, que capten la atención del usuario.

- **Implementación:** El formato de visualización debe ser robusto y consistente para todas las funciones que se implementen. Se debe realizar un diseño de la aplicación fácil de administrar y mantener.
- **Rendimiento.** Es la medida o cuantificación de la velocidad con que se realiza una tarea o proceso. El rendimiento de este software depende del tiempo de respuesta que tenga. La duración del tiempo de respuesta afecta la interacción del usuario con la aplicación, ya que, si éste es demasiado largo se podría perder la atención del tema estudiado.
- **Disponibilidad:** La aplicación debe estar disponible para cualquier usuario cuando este lo desee o lo considere necesario. En el caso de una versión para Internet, la disponibilidad dependerá del servidor, del estado de la conexión de red, entre otros.
- **Facilidad de aprendizaje:** Se deben incluir elementos didácticos que permitan facilitar el manejo del sistema, evitando que el tiempo de aprendizaje para la navegación e interacción con el software sea muy extenso.
- **Portabilidad:** El software debe ser independiente de la plataforma donde se ejecuta, y permitir su ejecución en varios sistemas operativos.
- **Fiabilidad:** El software debe garantizar al usuario una información veraz, así como, una ejecución y funcionamiento correcto, con baja probabilidad de fallas.

3.2.4.1 Requisitos Adicionales

Los requisitos adicionales son requisitos funcionales que intervienen en el desarrollo del software pero que no pueden ser asociados a un caso de uso en particular (podrían tener impacto en varios casos de uso o en ninguno).

A continuación, se muestra una lista con los requisitos adicionales considerados para el sistema. Estos restringen el desarrollo del proyecto, ya que, intervienen desde que se inicia su desarrollo, por lo cual, se deben tomar en cuenta para garantizar el buen funcionamiento de software.

Requisitos de plataforma de hardware

- Servidor:
 - Procesador Intel Pentium IV 750 MHz o superior.
 - 512 MB de memoria de acceso aleatorio (RAM).
 - Disco duro de 80 GB.
 - Conexión a Internet (preferiblemente).
 - Monitor a color con resolución de 1280x1624, preferiblemente de 17" (pulgadas).

- Cliente.
 - Procesador Intel Pentium.
 - 512 MB de memoria de acceso aleatorio (RAM).
 - Unidad de CD local.
 - Disponibilidad para conexión a Internet (preferiblemente).
 - Monitor a color con resolución de 1280x1624 o superior, preferiblemente de 17" (pulgadas).
 - Tarjeta gráfica.
 - Tarjeta de video.
 - Dispositivos de reproducción de audio y video digital.

Requisitos de plataforma de Software

- Servidor:
 - Sistema Operativo Windows 2000 / XP / Vista.

- Cliente:
 - Sistema Operativo Windows 2000 / XP / Vista.
 - Navegador WEB: Microsoft Internet Explorer / Mozilla FireFox.
 - Soporte para Macromedia Flash MX 2004 y DreamWeaver MX 2004.
 - Manejador de base de Datos My Sql (XAMP).

3.3 FLUJO DE TRABAJO: ANÁLISIS

En la fase de inicio del desarrollo del software, el flujo de trabajo análisis tiene como objetivos generales analizar requisitos, refinarlos y estructurarlos en un modelo de objetos que sirva como entrada al modelo de diseño. El resultado es un modelo inicial de análisis para definir con precisión los casos de uso y que sirva como guía en el establecimiento de la arquitectura candidata.

El modelo de análisis nos ayuda a refinar los requisitos obtenidos y nos permite establecer los aspectos internos del sistema, ya que, se representan mas especifica y formalmente los requisitos del sistema, mostrados previamente como casos de uso. Esto servirá de ayuda para guiarnos en el establecimiento de la arquitectura candidata.

El flujo de trabajo análisis, en la fase de inicio, incluye las siguientes actividades:

- *Analizar una clase y analizar un paquete:* Se describe de forma mínima las clases y los paquetes de análisis establecidos.
- *Analizar un caso de uso:* Se analizan y refinan los casos de uso escogidos detallando los más importantes. Se identifican las clases de análisis, se describen las interacciones entre objetos del análisis, se establecen las realizaciones de casos de uso y se capturan los requisitos especiales.
- *Análisis de la arquitectura:* Se identifican los paquetes de análisis y se identifican las clases de entidad, interfaz y control.

3.3.1 Identificación de Paquetes de Análisis.

Los paquetes de análisis proporcionan un medio para organizar los elementos del modelo de análisis en piezas manejables. Se identifica inicialmente como la forma de dividir el trabajo de análisis basándose en los requisitos funcionales y el dominio del sistema. El paquete de análisis puede constar de clases de análisis, realizaciones de casos de uso y de otros paquetes de análisis.

Los paquetes de análisis deben basarse en los requisitos funcionales del sistema, sin tomar en cuenta los requisitos no funcionales. Los paquetes de análisis tendrán una influencia fundamental en los subsistemas de diseño. Estos paquetes se convertirán, seguramente, en subsistemas de las 2 capas de aplicación superiores del modelo de diseño.

La Figura 3.19 muestra los paquetes de análisis identificados a partir de los casos de uso del SEMOI.

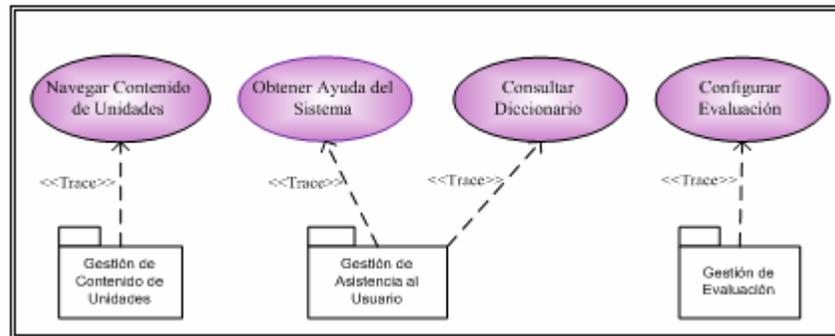


Fig. 3.19 Diagrama de Paquetes de Análisis

3.3.2 Identificación de Clases de Análisis.

Las clases, al igual que los demás elementos notacionales del UML, pueden estar clasificadas de acuerdo a varios criterios, como por ejemplo su objetivo dentro de un programa. Esta clasificación adicional se puede expresar mediante la utilización de estereotipos.

Los estereotipos más comunes utilizados para clasificar las clases son: Entidad Interfaz y Control. Cada estereotipo implica una semántica específica, lo cual constituye un método potente y consistente de describir las clases de análisis y contribuye a la creación de un modelo de objetos y arquitectura robustos.

- Clases de Interfaz: Se utiliza para modelar la interacción entre el sistema y el entorno del sistema (actores, sistemas externos). Esto implica recibir información y peticiones de usuario y los sistemas externos. Modelan las partes del sistema que dependen de sus actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema. Representan abstracciones de ventanas, formularios, paneles, interfaces de comunicación, etc.

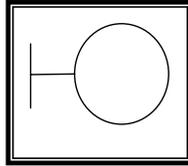
Representación:

Fig. 3.20 Clase de Interfaz

Para el caso de Uso Navegar Contenido de Unidades se considera la clase de Interfaz **IU de Navegabilidad** que muestra al usuario distintas opciones que presenta el software y proporciona acceso a ellas. Muestra los temas y la evaluación correspondiente a la unidad seleccionada.

- Clases de Entidad: Se utiliza para modelar información que posee una vida larga y que es a menudo persistente. Este tipo de clase suele reflejar entidades del mundo real o elementos necesarios para realizar tareas internas al sistema. También se denominan clase dominio, ya que suelen tratar con abstracciones de entidades del mundo real. Un objeto de entidad no ha de ser necesariamente pasivo y puede tener en ocasiones un comportamiento complejo y relativo a la información que representa. Los objetos entidad aíslan los cambios en la información que representan.

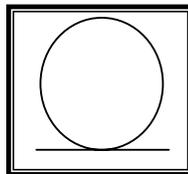
Representación:

Fig.3.21. Clase de Entidad

Para el SEMOI, se consideran las siguientes clases de entidad relacionadas al caso de uso Navegar contenido de Unidades:

Entidad Lección: Representa el contenido de cada de lección de la unidad explorada.

Entidad Evaluación: Herramienta que permite evaluar si el contenido de la asignatura a sido comprendido correctamente por el usuario durante el estudio.

Entidad Animación: Actividad didáctica que muestra los conceptos de cada unidad mediante imágenes con movimiento.

Entidad Juegos: Representa una actividad didáctica que complementa el contenido presentado en cada unidad utilizando juegos interactivos.

- **Clases de Control:** Modela el comportamiento secuenciado específico de uno o varios casos de uso. Se trata de clases que coordinan los eventos necesarios para llevar a cabo el comportamiento que se especifica en el caso de uso, representan su dinámica. Representan coordinación, secuencia, transacciones y control de los objetos y se usan para encapsular el control de un caso de uso en concreto.

Representación:

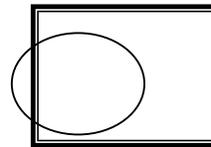


Fig.3.22. Clase de Control

Las clases de Control presentes en el SEMOI se describen a continuación:

Gestor de Navegabilidad: Permite verificar la opción seleccionada del usuario y ofrece acceso al micromundo correspondiente.

Gestor de Actividad Didáctica: Controla el acceso a cada una de las actividades didácticas seleccionadas por el usuario.

3.3.3 Realización de Casos de Uso-Análisis

Una Realización de Caso de Uso-Análisis es una colaboración dentro del modelo de análisis que describe como se lleva a cabo y se ejecuta un caso de uso determinado. Posee una descripción textual del flujo de sucesos, diagramas de clases que muestran sus clases de análisis participantes, y diagramas de interacción que muestran la realización de un flujo o escenario particular del caso de uso en términos de interacciones de objetos de análisis. [9]

3.3.3.1 Diagrama de Clases de Análisis

Una clase de análisis y sus objetos normalmente participan en varias realizaciones de casos de uso. Es importante durante el análisis coordinar todos los requisitos sobre una clase y sus objetos que pueden tener diferentes casos de uso. Para ello, se adjuntan diagramas de clases a las realizaciones de casos de uso, mostrando sus clases participantes y sus relaciones. El diagrama de la Figura 3.23 representa la realización del caso de uso Navegar por Contenido de Unidades.

Flujo de Sucesos – Análisis para el diagrama de clases de la realización del caso de uso Navegar por Contenido de Unidades

Como se muestra en la Fig. 3.23 el actor **usuario** utiliza la interfaz *IU de Navegabilidad* para acceder al contenido de las unidades. Esta interfaz se relaciona con *el Gestor de Navegabilidad* del sistema que permite al usuario explorar el contenido de la unidad requerida. A partir del *Gestor de Navegabilidad*, el usuario puede acceder a la *Lección*, *Evaluación* o al *Gestor de la actividad didáctica* correspondiente. El *Gestor de Actividad Didáctica* controla, a su vez, el acceso al *juego* o la *animación* de la unidad en estudio.

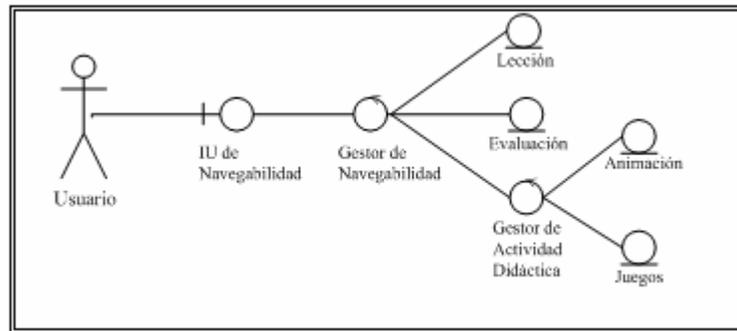


Fig. 3.23 Diagrama de clases de análisis de la realización del caso de uso Navegar por Contenido de Unidades.

3.3.3.2 Diagrama de Interacción

Los diagramas de interacción se utilizan para mostrar una interacción, que consiste de un conjunto de objetos y sus relaciones, incluyendo los mensajes que puedan ser realizados entre ellos. Modelan los aspectos dinámicos de un sistema, lo que conlleva a modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción muestran cómo se comunican los objetos.

Se deberían utilizar diagramas de interacción cuando se quiere analizar el comportamiento de varios objetos dentro del mismo caso de uso, resultando apropiados para mostrar colaboraciones entre objetos. Para ello, se prefiere utilizar un Diagrama de Colaboración, el cual, permitirá identificar los requisitos y responsabilidades sobre los objetos mostrando sus interacciones, creando enlaces entre ellos y añadiendo mensajes a esos enlaces.

de una animación (por medio del mensaje 6: *Observar*) o se puede realizar un juego interactivo (mediante 7: *Efectuar*) que complete la información percibida en el estudio de las lecciones correspondientes.

3.3.3.3 Descripción de la Aplicación

Para la realización de la aplicación se debe definir el tipo de material educativo computarizado que se utilizará. Los materiales educativos computarizados se clasifican en Heurísticos, Algorítmicos o mixtos que establece las diferencias entre dos enfoques educativos.

De acuerdo a las necesidades educativas de la población objetivo se determina que el software se desarrollara bajo un enfoque mixto, que posea características de los materiales heurísticos y de los algorítmicos.

Según la función que va a desempeñar el SEMOI, se establece como un programa tutorial con actividades interactivas que permitan el desarrollo y ejercitación del intelecto.

Este software educativo consta de 4 unidades, que abarcan todo el contenido de la asignatura Modelos de Operaciones I, presentando escenarios informativos, lúdicos, expresivos, evaluativos, interactivos y motivadores que le permitirán, al usuario, adquirir conocimientos que serán usados posteriormente.

La información será organizada por unidades, los cuales estarán divididos en lecciones para la mejor comprensión del usuario. Se incorporarán una serie de imágenes, juegos y animaciones que permitirán la interacción del usuario con la aplicación. Después de cada unidad se proporciona al estudiante la posibilidad de realizar una autoevaluación sobre el contenido dicho capítulo. Cada evaluación

constará de una serie de preguntas de selección, que deberán ser respondidas correctamente en un 70 % (como mínimo) para ser considerada aprobada.

3.3.3.4 Descripción de los Escenarios de Interacción

El software se apoya en distintos escenarios de interacción para lograr exponer la totalidad del contenido de la asignatura y completar el propósito del proceso enseñanza – aprendizaje. A continuación se muestra una breve descripción de estos escenarios:

- **Contenido Teórico:** Persigue que el estudiante logre interpretar y comprender la información de la asignatura, para ello se presentan conceptos básicos que permiten al estudiante obtener conocimiento específico del mismo.
- **Ejemplos Animados:** Se incorporan efectos animados a las unidades que faciliten la comprensión del contenido teórico.
- **Ejemplos Inanimados:** Se presentan imágenes estáticas que complementen el contenido teórico.
- **Evaluaciones:** Evaluaciones interactivas que permiten conocer el nivel de conocimiento que ha obtenido el estudiante durante cada lección.
- **Ayuda:** Permite que el usuario pueda obtener información sobre como indagar a través el software
- **Glosario:** Contiene palabras claves para el proceso de enseñanza de la asignatura. Permite al usuario obtener conceptos breves y precisos de diversos términos.

3.4 EVALUACIÓN DE LA FASE DE INICIO

Durante la Fase de Inicio de desarrollo del SEMOI, se trabajó con los flujos de trabajo Requisitos y Análisis. Se incorpora, al RUP, la metodología de software educativo de Álvaro Galvis, para la recolección de requisitos y el análisis preliminar de los componentes pedagógicos del sistema. Para obtener una visión general de todo el sistema, se necesitó solo una interacción de los flujos de trabajo fundamentales.

En el flujo de trabajo Requisitos se identificaron los principales requerimientos del software mediante un estudio de la población objetivo y del entorno donde ésta se desarrolla. Con base en estos requerimientos, se establece el Modelo de Casos de Uso inicial del sistema, los riesgos más importantes de la aplicación y la forma de mitigar cada uno de ellos.

En el flujo de trabajo Análisis, se esboza la arquitectura candidata del SEMOI. Esto se llevó a cabo, a través de la identificación de los Paquetes de Análisis que permitieron organizar mediante trazas los Casos de Uso identificados, y de la elaboración del Modelo de Clases de Análisis para la realización de los casos de usos identificados en el flujo de trabajo Requisitos.

3.5 PLANIFICACIÓN DE LA FASE DE ELABORACIÓN

Durante la Fase de Elaboración, se planea recopilar el resto de los requisitos del SEMOI, además se realizarán labores de análisis y diseño que permitan establecer la arquitectura base del sistema.

Se realizarán las interacciones necesarias de los flujos de trabajo implementación y/o pruebas (si es necesario), que permitan la definición de los

componentes principales del sistema y que ayude a robustecer la línea base de la arquitectura.

Además, corresponde a esta fase, el Diseño Educativo, donde se determinará el contenido educativo de la herramienta, estableciendo lo que se va a enseñar y las interfaces que transmitan de forma efectiva el conocimiento y que permitan mantener la motivación del usuario.

CAPÍTULO IV: FASE DE ELABORACIÓN

4.1 INTRODUCCIÓN

Durante la fase de inicio, se puso en marcha el proyecto identificando y analizando los requisitos, riesgos y la viabilidad del sistema, además, se analizaron las necesidades educativas de la población a la que está dirigido el software. Una vez establecidos los requerimientos del software y las necesidades de la población objetivo y la viabilidad del software para dar apoyo al sistema educativo actual, se debe establecer la línea base de la arquitectura, identificando la información o contenido que debe ofrecer la aplicación y las estrategias necesarias para la transmisión efectiva del conocimiento.

La fase de Elaboración se desarrolla mediante los flujos de trabajo requisitos, análisis, diseño y, en un menor grado, implementación.

Durante el flujo de trabajo Requisitos, se recopilan parte de los requisitos que podrían haber quedado pendientes en la fase de inicio, lo que permitirá, especificar con más detalle los casos de uso y establecer las interfaces de usuario.

En el flujo de trabajo Análisis, se refinan y analizan detalladamente los nuevos casos de uso, clases y paquetes más críticos identificados, proporcionando una visión general y mejor comprensión del sistema. Posteriormente, se realizará el análisis del contenido programático de la asignatura.

En el flujo de trabajo Diseño se modela el sistema, tomando en cuenta los requisitos funcionales y no funcionales del software, así como, las restricciones establecidas. Se identifican las clases que conformarán cada módulo del software y

se establece el diseño de la arquitectura. Se realizará el diseño de la interfaz de usuario. Con base en el de análisis educativo de la asignatura y de su contenido programático, se procederá a realizar el diseño educativo, a través del cual, se establecerá el contenido educativo de la aplicación y las características que, de acuerdo a los requerimientos del usuario, debe tener la interfaz para mostrar dicho contenido.

Finalmente, para el flujo de trabajo implementación, podrían crearse prototipos de la interfaz de usuario, dependiendo de las necesidades del diseño.

4.2 FLUJO DE TRABAJO: REQUISITOS

En la fase de inicio se capturan la mayoría de los requisitos fundamentales de la aplicación y, en base a ello, se realiza el análisis educativo, sin embargo, para llevar a cabo el diseño educativo, se deben considerar nuevos requerimientos del software y la construcción de un modelo de casos de uso mas completo que refleje las características del software, según las nuevas necesidades de la población objetivo.

4.2.1 Representación De Los Requisitos Como Casos De Uso

Para la fase de Elaboración, se realiza una representación mas completa de los requisitos del usuario, a través de un nuevo modelo de casos de uso, para ello se toma como base el diagrama de casos de uso de la fase anterior (Ver Figura 3.18).

Actores

Para el nuevo modelo se mantiene los mismos usuarios establecidos previamente, sin embargo, se hace necesaria la creación del actor Asistente de Usuario, que representa la documentación que proporcionará asistencia o ayuda al usuario. Este actor se

relacionará con los casos de uso Obtener Ayuda del Sistema y Consultar Glosario. Los actores usuario, administrador y base de datos SEMOI conservan sus mismas funciones, es decir, interactúan con los mismos casos de uso establecidos. Sin embargo, la inclusión de nuevos casos de uso especifica nuevas funcionalidades del sistema a las que tienen acceso los actores.

Casos de Uso

En la tabla 4.1 se observan los casos de uso incluidos al nuevo modelo de casos de uso. Como se menciona anteriormente, los nuevos casos de uso permiten extender y especificar más la funcionalidad de la aplicación.

Tabla 4.1. Nuevos Casos de Uso (1/2)

NOMBRE	DESCRIPCIÓN
Cambiar Contraseña	Caso de uso que permite al usuario, modificar la contraseña actual.
Configurar Sistema	Caso de uso empleado para proporcionar, al administrador, la opción modificar datos de la aplicación (incluye información del usuario y de la evaluación).
Listar Alumnos	Proporciona, al administrador, un listado completo de todos los alumnos registrados en el sistema
Imprimir Listado de Alumnos	Caso de uso que permite al administrador imprimir un listado de los alumnos registrados en el sistema.

Tabla 4.1. Nuevos Casos de Uso (2/2)

NOMBRE	DESCRIPCIÓN
Configurar Datos de Usuario	Este caso de uso permite la configuración de los usuarios. Incluye la inserción, actualización y eliminación de usuarios del sistema. El administrador es el encargado de estas tareas.
Registrar Usuario	Caso de uso que gestiona la inserción de usuarios en el sistema.
Actualizar Usuario	Caso de uso que gestiona la actualización de datos de los usuarios previamente registrados en el sistema.
Eliminar Usuario	Caso de uso que gestiona la eliminación de usuarios en el sistema
Modificar	Permite, al administrador del sistema, modificar preguntas y/o respuestas de alguna evaluación.
Consultar Glosario	Caso de uso que sustituye al anterior “Consultar diccionario”, y proporciona un glosario de términos referentes a las unidades de estudio.

Adicionalmente, se ha depurado el diagrama eliminado los casos de uso efectuar juegos y observar animación. Por otra parte, se han reestructurado las relaciones entre algunos casos de uso y los actores del sistema (Ver Fig. 4.1).

Diagrama de Casos de Uso

En el diagrama de la Figura 4.1 se puede observar las modificaciones realizadas a los casos de uso.

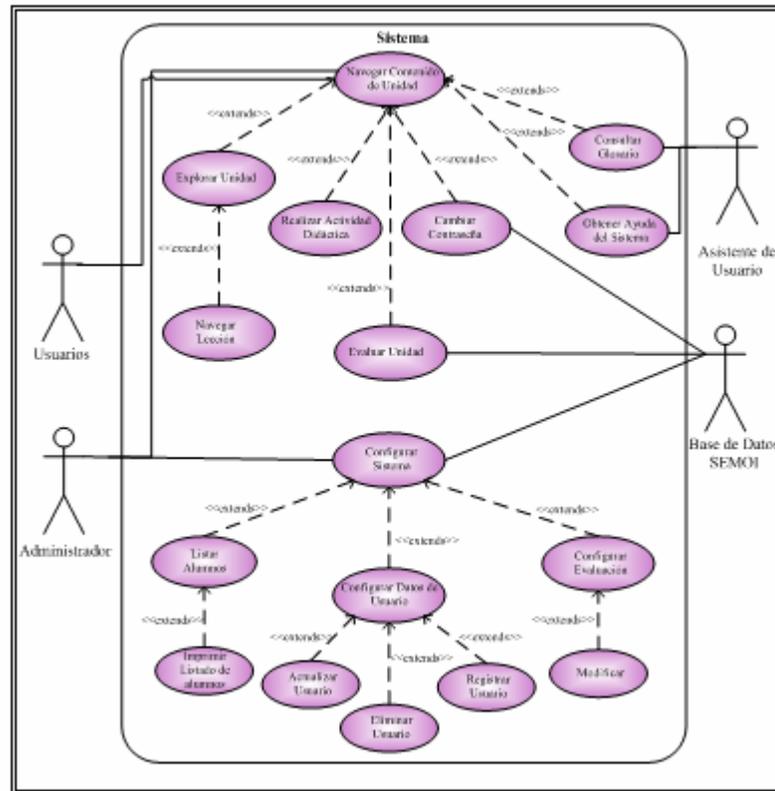


Fig. 4.1. Diagrama de Casos de Uso del SEMOI

La tarea fundamental del sistema es permitir al actor “**usuario**” *Navegar por el Contenido de Unidades*, por lo cual se debe proveer una herramienta que proporcione facilidad al estudiante. El caso de uso *Navegar por el Contenido de Unidades* le ofrecerá al usuario la opción de explorar una unidad de estudio (*Explorar Unidad*) y, a su vez, una lección específica de esa unidad (*Navegar lección*), cuando así lo desee. De forma paralela, el estudiante podrá realizar el cambio de su contraseña si así lo desea (*Cambiar Contraseña*). Además, podrá realizar, la correspondiente actividad didáctica, representado por el caso de uso *Realizar Actividad Didáctica*, y, finalmente, realizará la evaluación de la unidad estudiada (*Evaluar Unidad*). Tanto para el cambio de la contraseña del usuario, como para la evaluación de la unidad, se requiere que el sistema consulte de la base de datos la información referente a las

evaluaciones y al proceso de login del usuario. Adicionalmente, el usuario puede *consultar el glosario* y la *ayuda* del sistema mientras estudia las unidades del software.

En el diagrama de casos de uso, igualmente, se puede observar que, para el actor “**administrador**”, se especifican las funciones del caso de uso *Configurar Sistema*. Una vez que el administrador requiera *Configurar el Sistema* tendrá las opciones de *Listar los Alumnos Registrados*, así como de *imprimirlos* mediante una impresora; también podrá *Configurar Datos del Usuario*, lo que a su vez, le permitirá *Registrar, Actualizar y/o Eliminar usuarios*; igualmente tendrá la posibilidad de *configurar la evaluación* de cada unidad, modificando las preguntas y/o respuestas previamente establecidas por el administrador. Para ello, se deberá consultar a la base de datos existente y posteriormente, se realizarán las modificaciones deseadas. Esto permitirá mantener actualizados usuarios y evaluaciones.

4.3 FLUJO DE TRABAJO: ANÁLISIS

En el flujo de trabajo requisitos se refino el modelo de casos y se estructuraron los requisitos en un modelo de objetos que sirve. Al obtener un diagrama de casos de uso mas completo se profundiza el sistema y sus procesos como entrada al modelo de diseño.

En este flujo de trabajo serán analizados en profundidad los Casos de Uso, permitiendo establecer los aspectos internos del sistema. Esto servirá de ayuda para guiarnos en el establecimiento de la arquitectura candidata.

El análisis se realiza a través de la identificación de paquetes y clases de análisis, basándose en los nuevos casos de uso.

4.3.1 Identificación de Paquetes de Análisis

El diagrama de paquetes de análisis permite la representación del sistema en piezas manejables. Los diagramas de paquetes de análisis suministran una descomposición de la jerarquía lógica de un sistema.

En la Figura 4.2 se observa la identificación de los paquetes de análisis a partir del modelo de casos de uso redefinido en esta fase. Se incluyen ciertas modificaciones al diagrama de la Fig. 3.19, realizado en la fase de inicio.

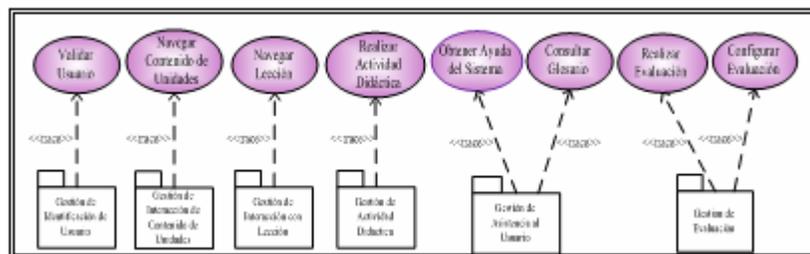


Fig.4.2 Diagrama de Paquetes de Análisis del SEMOI

Se incluyen los paquetes de análisis Gestión de Identificación de Usuario, que corresponde al proceso de login del software; Gestión de Interacción con Lecciones, que incluye todas las lecciones a las que el usuario puede acceder; Gestión de Actividad Didáctica y Gestión de Asistencia al Usuario. Esta reestructuración permite mantener mejor organizados los casos de uso del sistema

A partir del bosquejo de paquetes realizado al comienzo de cada iteración, principalmente considerando los servicios del sistema y el agrupamiento de clases en paquetes, se refinan y mantienen los mismos durante toda la Fase.

4.3.2 Identificación de Clases de Análisis.

Las clases de análisis permiten la representación de abstracciones de clases y subsistemas del diseño del software. Dentro del modelo de análisis del sistema los casos de uso se describen mediante clases de análisis y sus objetos.

Al incluir nuevos casos de uso, es necesario distinguir nuevas clases de entidad, interfaz y control para la representación de las clases de análisis. Además, no solo se realizará el análisis del caso de uso *Navegar Contenido de Unidades*, sino que se analizará, igualmente, el caso de uso *Configurar Sistema*.

- **Clases de Interfaz:** Para el caso de uso Navegar Contenido de Unidades se mantiene la interfaz *IU de Navegabilidad*. Para el caso de uso Configurar Sistema se requiere de la Interfaz *IU de Configuración de Sistema*, que permite al administrador interactuar con el sistema para realizar labores de visualización, borrado, actualización y/o inserción de usuarios, así como, de preguntas o respuestas en las evaluaciones, según se requiera.
- **Clases de Entidad:** Para el caso de uso Navegar Contenido de Unidades se incluye las siguientes entidades: *unidad*, que almacena y proporciona al usuario la información referente a la unidad explorada; *usuarios*, que representa a todos los usuarios registrados en el sistema; por su parte, las entidades *animación* y *juegos* creadas anteriormente, son sustituidas por una unidad denominada *Actividad Didáctica*; finalmente, se agregan las entidades *Ayuda y Glosario*, que proporcionarán asistencia al usuario. En el caso de uso Configurar Sistema, se distingue igualmente la entidad *usuarios* y la clase de entidad *Evaluación*, donde se almacenan las evaluaciones de cada unidad de estudio.

- **Clases de Control:** El *Gestor de Navegabilidad* presentado para la fase de inicio en el análisis del caso de Uso Navegar Contenido de Unidades se renombra como *Gestor de Unidades* para la nueva representación, ya que, resulta ser una denominación más conveniente para la identificación de sus funciones. Igualmente, se agregan a partir del gestor de unidades, las clases de control: *Gestor Lección*, que permite al usuario navegar la lección correspondiente a la unidad estudiada; *Gestor de Cambio de Contraseña*, el cual, permite procesar la solicitud del usuario de modificar su contraseña; *Gestor de Validación de Datos de Usuario*, que permite validar los datos del usuario que solicita el cambio de contraseña; Gestor de Actualización de Datos, gestiona la actualización de la contraseña del usuario; Gestor de Evaluación, que procesa la activación de la evacuación; el *Gestor de Procesamiento de Respuestas*, procesa las respuestas de los usuarios validando si son correctas o no; *Gestor de Ayuda* que proporciona al usuario la posibilidad de obtener la ayuda del sistema; y *Gestor de Glosario* que muestra al usuario un conjunto de términos correspondientes a las unidades de estudio. Para el caso de uso Configurar Sistema, se identifican las clases de control *Gestor de Configuración del sistema*, que permite procesar las solicitudes del administrador de configurar usuarios y evaluaciones; *Gestor de Listado de Alumnos*, proporcionan un listado de los usuarios (alumnos o estudiantes) registrados en el sistema; *Gestor de impresión*, procesa la impresión del listado de alumnos registrados en el sistema mediante una impresora previamente instalada y configurada; *Gestor de Configuración de Usuario*, que procesa las solicitudes de inserción, actualización y/o eliminación de usuarios; *Gestor de Registro*, gestiona el registro de usuarios en el sistema, *Gestor de Actualización*, procesa la solicitud de actualización de los datos de un usuario registrado en el sistema; *Gestor de Eliminación*,

que procesa la solicitud de eliminación de uno o todos los usuarios de la base de datos; *Gestor de Validación* de datos de usuarios, que verifica los datos de los usuarios que serán eliminados o cuyos datos serán modificados; *Gestor de Procesamiento de Usuarios*, actualiza los usuarios gestionando los cambios realizados, ya sea, modificación o eliminación; *Gestor de Configuración de evaluación*, que permite al administrador actualizar las evaluaciones ofreciendo acceso a la opción de modificación de preguntas y/o respuestas en las evaluaciones; *Gestor de modificación de evaluación*, que permite la modificación de preguntas y/o respuestas en las evaluaciones.

4.3.3 Realización de Casos de Uso-Análisis

Una realización de caso de uso se describe en términos de clases de análisis y de sus objetos, centrándose en los requisitos funcionales del sistema. Los requisitos funcionales se representan a través el modelo de caso de usos del sistema.

Debido a que, para fase de elaboración, se agregan nuevos casos de uso al modelo original, es necesaria la identificación de la realización de casos de uso correspondiente. Esto nos lleva al establecimiento de los diagramas de clases de análisis y de interacción, que permitan mostrar plenamente como se lleva a cabo la interacción entre los objetos para cada caso de uso.

4.3.3.1 Diagramas de Clases de Análisis

Los diagramas de clases de análisis permiten representar la realización de los casos de uso del sistema.

El diagrama de la Figura 4.3 representa la realización del caso de uso *Navegar Contenido de Unidades*.

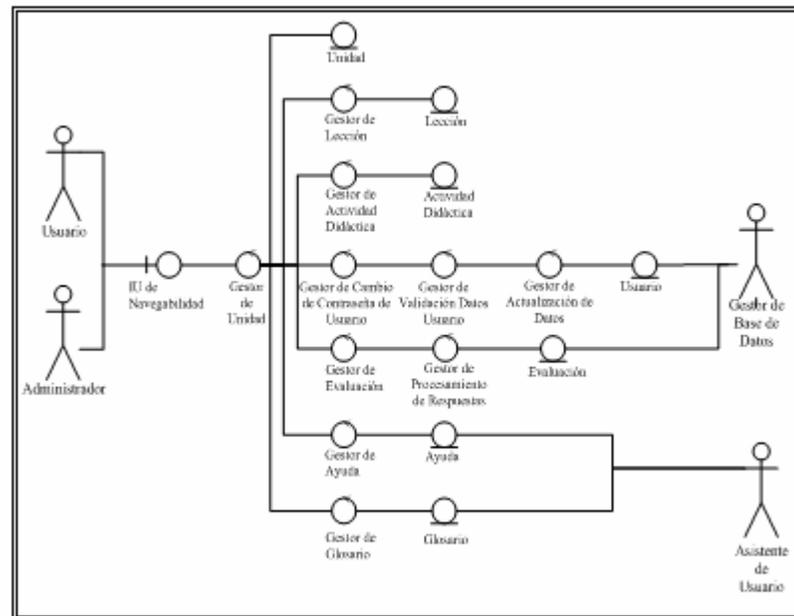


Fig.4.3. Diagrama de clases de análisis de la realización del caso de uso Navegar Contenido de Unidades

A través de la **IU de Navegabilidad**, tanto el administrador como el usuario interactúan con la aplicación y acceden al menú principal. **El Gestor de Unidad** permite seleccionar y explorar una **Unidad** de estudio del software, a su vez, pueden escoger una **Lección** específica, mediante el **Gestor de Lección**. A través de los **Gestores de Ayuda y Glosario** se permite consultar el glosario y la ayuda del sistema, respectivamente. Por otra parte, el **Gestor de Unidad** permite la selección y ejecución de la **Actividad Didáctica**, de la **Evaluación** correspondiente a cada **Unidad**, y del **Cambio de Contraseña**. Para ello, se requiere del Gestor de Actividad Didáctica, del **Gestor de Evaluación** y del **Gestor de Cambio de Contraseña**, respectivamente. **El Gestor de Evaluación** necesita de un **Gestor de Procesamiento de Respuesta** que permite la verificación de la respuesta seleccionada por el usuario. El **Gestor de Cambio de Contraseña** requiere, a su vez, de un **Gestor de Verificación de Los**

Datos del Usuario y de un *Gestor de Procesamiento de Datos*, que permitan completar el proceso de actualización de la contraseña.

Para el caso de uso *Configurar Sistema* se desarrolla el diagrama de clases de análisis de la Figura 4.4. El administrador interactúa con la aplicación a través de la *IU Configuración del Sistema*, permitiendo el acceso a las opciones de configuración de usuarios y de evaluaciones. Específicamente, el *Gestor de Configuración de Sistema*, permite al administrador realizar las tareas de *Listado, Borrado, Inserción, y Modificación de Usuarios* en el sistema. Adicionalmente, proporciona la opción de modificación o actualización de preguntas y/o respuestas en las *Evaluaciones* de cada unidad.

4.3.3.2 Diagramas de Interacción

La interacción entre los objetos se representa mediante diagramas de colaboración. Estos diagramas muestran, a través de mensajes, como se comunican los objetos.

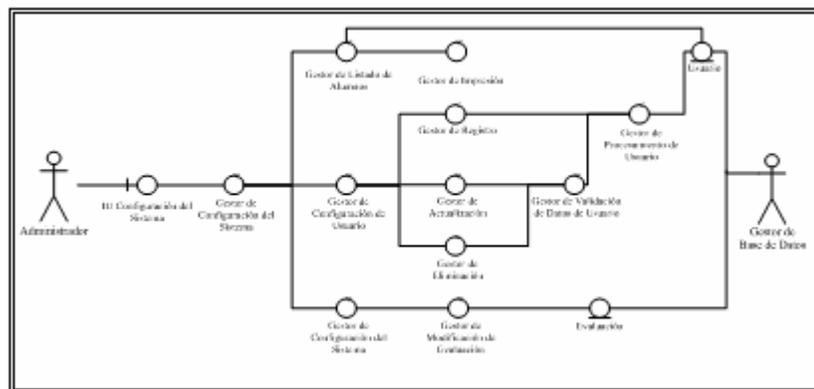


Fig.4.4. Diagrama de clases de análisis de la realización del caso de uso Configurar Sistema.

En la Figura 4.5 se puede observar la colaboración de objetos en la realización del caso de uso Navegar Contenido de Unidades.

Flujo de Sucesos – Análisis para el diagrama de colaboración de la realización del caso de uso Navegar Contenido de Unidades

En el diagrama de colaboración de la Fig. 4.5, se muestra el proceso requerido para que el usuario y el administrador naveguen por el contenido de las unidades de la asignatura Modelos de Operaciones I en el SEMOI. A través de la IU de Navegabilidad se muestran las opciones que permiten al usuario seleccionar una unidad de estudio, esto se evidencia en el diagrama, mediante el mensaje **1: Unidad Seleccionada**, posteriormente, el mensaje 2 se utiliza para enviar al Gestor De Unidad la información de la unidad seleccionada por el usuario. Una vez escogida la unidad de estudio, el Gestor de Unidad, permite al usuario, explorar una Unidad (a través del mensaje **3: Obtener Información de Unidad Seleccionada**). Cada Unidad está compuesta por un conjunto de Lecciones, a las que se dará el acceso igualmente, mediante la IU de Navegabilidad, esto se puede apreciar en el diagrama a través del mensaje **4: Lección Seleccionada**, posteriormente, esta opción se envía al Gestor De Unidades mediante el mensaje **5: Enviar Opción de Lección Seleccionada**. A través del mensaje **6: Procesar Opción de Lección Seleccionada** se proporciona, al Gestor de Lección, la información referente a la lección de la unidad que se va a estudiar, para culminar este proceso se obtienen los datos de la lección seleccionada de la entidad Lección (**7: Obtener Información de Lección**). El Gestor de unidades permite también el acceso a la actividad didáctica enviando el mensaje **8: Seleccionar Opción de Actividad Didáctica**, hasta el gestor de actividad didáctica. A partir de este gestor, se obtiene mediante el mensaje **9: Capturar Actividad Didáctica** la actividad, animación o juego correspondiente a esa unidad. Para el cambio de contraseña de usuario, se envía el mensaje **10: Activar cambio de contraseña**, desde el Gestor de Unidades hasta el Gestor de Cambio de Contraseña de Usuario; para ello,

se requiere de la validación de los datos del usuario, para lo cual, se remite el mensaje **11: Validar Datos de Usuarios**, hasta el Gestor de Validación de Datos de Usuario, del cual se envía el mensaje **12: Procesar Datos de Usuarios**, hasta el Gestor de Procesamiento de Nuevos Usuarios, los datos actualizados se almacenan en la base datos, esto es a través de los mensajes **13: Almacenar datos de Usuarios** y **14: Manejador de Datos de Usuario**. El mensaje **15: Activar Evaluación** es enviado desde el Gestor de Unidades hasta el Gestor de Evaluación, cuando el usuario desee realizar la evaluación, para lo cual deberá haber leído todas las lecciones de la unidad estudiada. A partir de este Gestor, se utiliza el mensaje **16: Procesar datos de Evaluación** para conceder al Gestor de Procesamiento de Respuestas el trámite de las respuestas seleccionadas por el usuario en la evaluación. Luego, se verifican las respuestas enviando, a la entidad Evaluación el mensaje **17: Verificación de Respuestas**, para lo cual se consulta a la base de datos a través del mensaje **18: Manejador de datos usuario**. Para la consulta de la ayuda del sistema se envía desde el Gestor de Unidades el mensaje **19: Solicitar ayuda** y luego a través del mensaje **20: Mostrar Ayuda** se proporciona al usuario o al administrador la ayuda solicitada. Finalmente, partiendo del Gestor de Unidades, se puede consultar el glosario enviando el mensaje **21: Solicitar Glosario hasta el Gestor de Glosario**, mientras que, a través del mensaje **22: Mostrar Glosario** se obtienen los datos recopilados en el glosario. Para obtener los datos de ayuda y de glosario, la entidad que corresponde a cada una de ellos debe enviar el mensaje **23: Manejador de Datos de Asistencia de Usuario** al actor Asistente de Usuario que le proporcionará la información requerida.

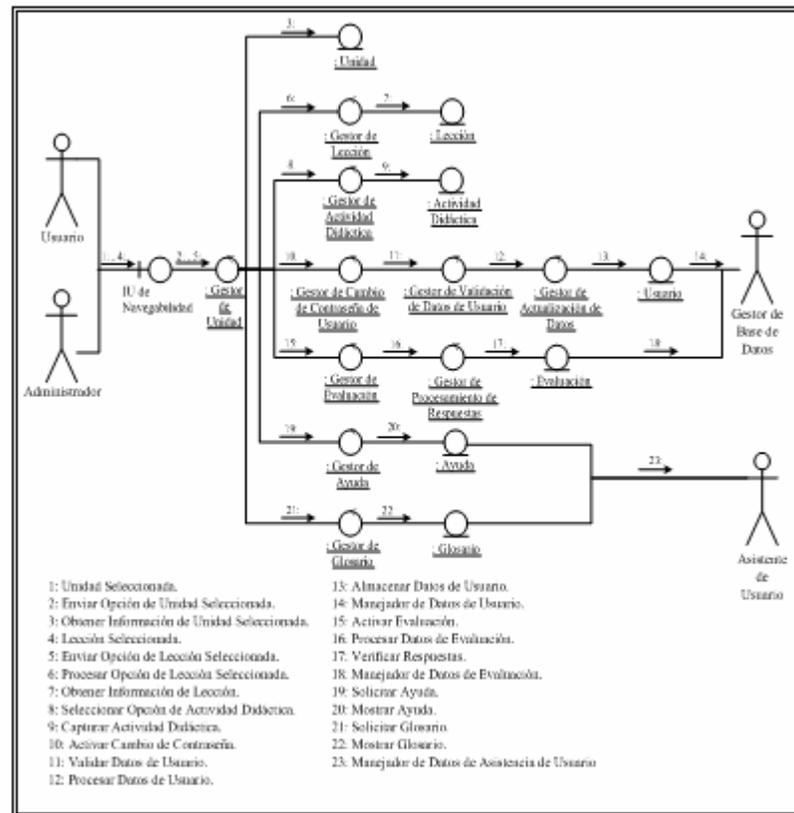


Fig.4.5. Diagrama de Colaboración de la realización del caso de uso Navegar Contenido de Unidades.

El diagrama de colaboración para la realización del caso de uso *Configurar Sistema*, se muestra en la Figura 4.6.

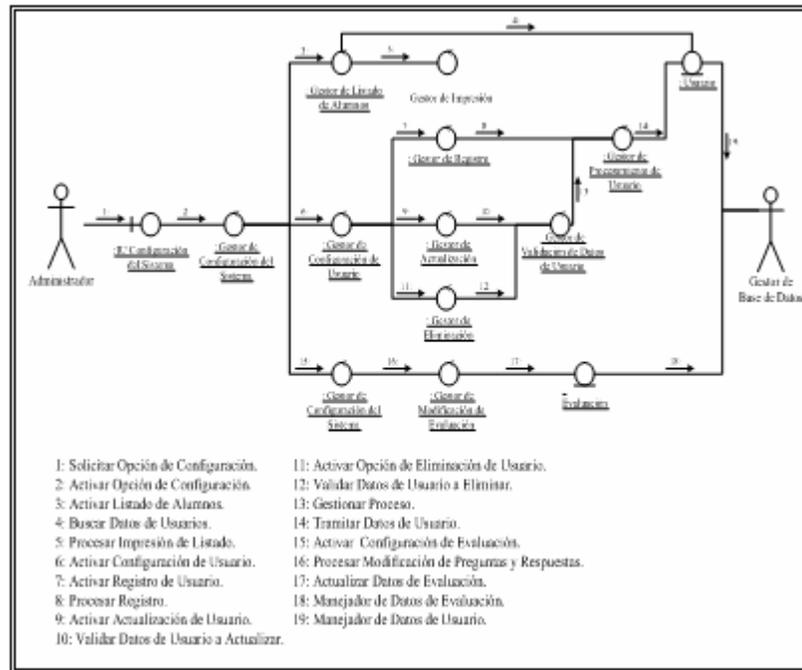


Fig.4.6. Diagrama de Colaboración de la realización del caso de uso Configurar Sistema

Flujo de Sucesos – Análisis para el diagrama de colaboración de la realización del caso de uso Configurar Sistema

En la Fig. 4.6 Se observa el diagrama de colaboración del caso de uso Configurar Sistema, donde el Administrador inicialmente, interactúa con la aplicación a través de la IU Configuración de Sistema. A través, de esta interfaz el Administrador solicita, observa y explora las opciones de configuración de usuarios y evaluaciones (**1: Solicitar Opción de configuración**). Mediante, **2: Activar Opción de Configuración**, el administrador accede al Gestor de Configuración de Sistema, indicando la opción a ejecutar. El Gestor de Configuración del Sistema controla los procesos de listado, borrado, modificación e inserción de usuarios. Para la primera tarea, envía al Gestor de Listado de Alumnos, el mensaje **3: Activar Listado de Alumnos**, cuando el

administrador requiera la lista de los alumnos registrados; mediante el mensaje **4: *Buscar Datos de Usuarios*** y **19: *Manejador de Datos de Usuarios*** se obtienen los datos de todos los usuarios y se muestran en pantalla. Si el administrador del sistema requiere la impresión de un reporte para el listado de alumno, se envía desde el Gestor de Listado de Alumnos, el mensaje **5: *Procesar Impresión de Listado***, hasta el Gestor de Impresión, el cual, a su vez, emplea el mensaje 19 para culminar el proceso de impresión del listado. Para configuraciones de usuario, el Gestor de Configuración del Sistema remite el mensaje **6: *Activar Configuración de Usuarios*** hasta el Gestor de Configuración de Usuario. Desde este gestor se requiere del mensaje **7: *Activar Registro de Usuario*** para permitir al Gestor de Registro de Usuario insertar uno nuevo, luego se envía el mensaje **8: *Procesar Registro***, al Gestor de Procesamiento de Usuario, y, por ultimo, se emiten los mensajes **14: *Tramitar datos de Usuario*** y **19** para almacenar los datos del usuario. Para la actualización de Usuarios, el Gestor de Configuración del Sistema se comunica con el Gestor de Actualización utilizando el mensaje **9: *Activar Actualización de Usuario***. Este Gestor se comunica, a su vez, con el Gestor de Validación de Usuario mediante el mensaje **10: *Validar Datos de Usuario a Actualizar***, para verificar la identidad del usuario que se desea actualizar. A través del mensaje **13: *Gestionar Proceso*** se dirige hacia el Gestor de Procesamiento de Usuario e igualmente se tramitan y se manejan los datos del usuario utilizando los mensajes **14** y **19**, mencionados anteriormente. Para el borrado de usuarios, se establece la comunicación entre el Gestor de Configuración del Sistema y el Gestor de Eliminación mediante el mensaje **11: *Activación de Opción de Eliminación de Usuario***. Este Gestor de Eliminación envía el mensaje **11: *Validar datos de Usuarios*** a Eliminar al Gestor de Validación de Datos de Usuario para comprobar datos del usuario a eliminar, posteriormente, se procesa la eliminación enviando el mensaje **13: *Gestionar proceso*** al Gestor de procesamiento de usuario. Para completar el borrado se utilizan igualmente los mensajes **14** y **19** para el trámite final en la base de datos. Si el administrador selecciona la opción de configurar alguna de las evaluaciones, el Gestor de Configuración del Sistema envía al Gestor de

Configuración de Evaluación, el mensaje **15: Activar Configuración de Evaluación**, luego este Gestor de Evaluación, debe enviar un respectivo mensaje al Gestor de Modificación de Evaluación (**16: Procesar Modificación de Preguntas y Respuestas**) que permita realizar la actualización a la pregunta seleccionada. El mensaje **17: Actualizar Datos de Evaluación** es enviado a la entidad evaluación para modificar los datos almacenados, para ello, se emplea, finalmente, el mensaje **18: Manejador de Datos de Evaluación**, que representa la conexión y consultas con la tabla evaluación de la base de datos del SEMOI.

4.4 FLUJO DE TRABAJO: DISEÑO

En el diseño se modela el sistema y se establece la arquitectura estable y sólida que soporte los requisitos analizados previamente. Se crea un punto de partida para las actividades de implementación diseñando interfaces, clases y paquetes.

Para la fase de elaboración se realizará el diseño de la interfaz con el usuario, el diseño de la estructura del software, y el diseño de la base de datos.

Para el diseño de la interfaz con el usuario se utiliza la metodología de Álvaro Galvis, referente al desarrollo de MECs. Se definirán claramente los elementos que se determinaron como necesarios en todo micromundo interactivo y aquellos deseables que convenga para el caso. A esta fase corresponde: el Diseño Educativo, donde se establece lo que se va a enseñar y de que manera se impartirán esos conocimientos (incluyendo estrategias de motivación, evaluación, etc.), el Diseño Comunicacional, donde se identifica la estructura, características y componentes de las interfaces de usuario y el Diseño Computacional que engloba el diseño de la estructura del software.

El diseño de la estructura del software permite especificar clases, capas, paquetes e interacción entre los objetos del sistema, y a su vez, proporciona una arquitectura consistente para la fase de construcción. Para ello, se desarrolla el modelo de diseño del sistema, modelo de navegación, el diagrama de secuencia correspondiente, y el diagrama de capas de la aplicación. Esto se incluirá en el diseño computacional del software, ya que, forma parte del diseño de clases y objetos que serán implementados posteriormente.

Finalmente, se diseñará la base de datos del sistema, donde se almacenarán los usuarios del sistema, y las preguntas y respuestas de cada evaluación.

4.4.1 Diseño de la Interfaz

La interfaz de usuario permite el flujo de información entre un usuario y la aplicación. El diseño de la interfaz de usuario para el SEMOI se basa en los elementos descritos por Galvis, quien define 3 etapas para el desarrollo de un software educativo: diseño educativo, diseño comunicacional y diseño computacional. En esta fase se establecerá el diseño educativo, el diseño comunicacional y, parte, del diseño computacional.

4.4.1.1 Diseño Educativo del SEMOI

El diseño educativo de un MEC permite establecer el contenido y la estructura de la aplicación basándose en los requisitos recolectados previamente, y utilizando las técnicas y estrategias más adecuadas para la presentación de la información, que favorezcan el aprendizaje de la asignatura. Esto permitirá, a su vez, la realización de un bosquejo adecuado de la interfaz de usuario.

El diseño educativo pretende responder las siguientes interrogantes: ¿Qué enseñar con el MEC?, ¿En que ambiente o micromundo aprender?, ¿Cómo motivar y mantener motivados a los usuarios?, ¿Cómo saber que el aprendizaje se está logrando?

¿Qué enseñar?

El contenido programático de la asignatura Modelos de Operaciones I

El diseño educativo depende directamente de los resultados que se obtienen de la fase de análisis. En esta etapa se define la orientación y el contenido que tendrá el software, a través de la planificación del contenido del curso, las actividades diseñadas para incentivar el desarrollo de las capacidades del aprendizaje.

El contenido programático de la asignatura Modelos de Operaciones I que rige el Departamento de Computación y Sistemas, (Ver Anexo B), está dividido en cuatro unidades las cuales serán contempladas en su totalidad por el SEMOI. Dicho contenido actúa como guía para la formulación del objetivo general y los objetivos específicos del software educativo basándose en los objetivos y contenido de cada unidad.

Para entender mejor este apartado se expone a continuación, la definición de algunos términos que se emplean, los cuales permitirán la mejor comprensión del punto que se está tratando.

Objetivos generales: son enunciados que representan los comportamientos más complejos y los contenidos más amplios que se pretende que logren los estudiantes al finalizar el programa.

Objetivo terminal: es un enunciado descriptivo y cuantificable de lo que se ha de demostrar cuando se haya logrado el aprendizaje contemplado en el diseño. Se anuncia en relación a determinados indicadores y en función de las siguientes preguntas: ¿Qué debe hacerse? ¿Quién debe hacerlo? ¿Bajo que condiciones? ¿Cuáles son los criterios?

Conducta de entrada: Conducta o instrucción que debe poseer y mostrar el estudiante cuando ingresa a un curso o asignatura o unidad de aprendizaje. Abarca información referente al rendimiento escolar, tipo de instrucción y adiestramiento recibido. Algunas veces incluye, si es pertinente, informaciones tales como status socioeconómico, antecedentes físico-psicológicos que puedan ayudar al desarrollo de la instrucción.

Objetivos específicos: Enunciado que especifica exactamente la conducta a ser exhibida, las condiciones bajo las cuales tendrá lugar y el patrón de rendimiento mínimo aceptable.

La metodología de Galvis, establece en el diseño educativo, que a partir de las necesidades que se deseen atender con el MEC, se deriva el objetivo terminal que deberá poder alcanzar quien lo estudie. La redacción de los objetivos generales y específicos debe estar lo más sencillo posible, es decir, tienen que redactarse en términos operacionales, reflejando que se quiere alcanzar con el uso del material.

Con respecto a la especificación de los conocimientos previos, Galvis plantea, que las habilidades y destrezas que debe tener el usuario son los que finalmente van a determinar el éxito o no del material educativo computarizado o en todo caso le hace el camino más difícil al mismo.

A continuación, se enuncia el objetivo general del SEMOI:

Objetivo General

“Apoyar el proceso de enseñanza y aprendizaje para la optimización de sistemas a través de la enunciación y publicación de información asociada a modelos de programación matemática”.

Mediante la siguiente tabla (4.2) se puede apreciar los objetivos generales formulados para cada unidad exponiendo a su vez las lecciones que corresponde a cada una.

Tabla 4.2. Objetivos generales y lecciones de SEMOI. (1/2)

UNIDADES	OBJETIVO GENERAL	LECCIONES
I Análisis de Redes Pert-CPM.	Recordar la importancia del análisis de redes PERT CPM	Representación de la red
		Método de revisión hacia adelante
		Método de revisión hacia atrás
		Redes con incertidumbre
		Análisis de redes considerando costos
I Teoría de Juegos.	Reconocer la teoría de juegos para la toma de decisiones.	Formulación de juegos de 2 personas
		Formulación de juegos con estrategias mixtas

Tabla 4.2. Objetivos generales y lecciones de SEMOI. (2/2)

UNIDADES	OBJETIVO GENERAL	LECCIONES
III Programación	Identificar el modelo en reversa y en avance para la	Recursión en reversa y en avance
		Modelo de la ruta mas corta

Dinámica.	optimización de sistemas.	Modelo de tamaño de fuerza de trabajo
		Aplicaciones de la programación dinámica
IV Programación No Lineal.	Reconocer los problemas no lineales de optimización.	Función convexa y cóncava
		Optimización no restringida
		Optimización restringida
		Método de multiplicadores de Lagrange

Posteriormente, se publica de forma más detallada la redacción de objetivos para cada una de las lecciones correspondientes a las unidades del SEMOI. La formulación de objetivos terminales, conducta de entrada y objetivos específicos se realiza en base al programa de la asignatura Modelos de Operaciones I (anexo A), donde se describen ciertas destrezas o habilidades que el estudiante obtendrá una vez terminada la unidad de estudio. Los objetivos del software fueron manifestados de manera sencilla, ya que, el programador no puede tener plena seguridad de que el usuario este adquiriendo un aprendizaje total del contenido del tema.

Por razones de demostración, en este apartado solo se revelarán los objetivos instruccionales de las siguientes lecciones: Representación de la red, método de revisión hacia adelante, método de revisión hacia atrás, redes con incertidumbre y análisis de redes considerando costos, todas pertenecientes a la primera unidad **“Análisis de las Redes PERT-CPM”**. Sin embargo, todas las formulaciones antes expuestas fueron realizadas para el resto de las unidades que abarca el SEMOI: Programación no lineal, programación dinámica y teoría de juegos.

Unidad I: Análisis de Redes Pert - CPM

Lección 1: Representación de La Red.

Objetivos Terminales:

1. Ofrecer las herramientas necesarias para que el estudiante comprenda problemas relacionados a la investigación de operaciones.
2. Planificar actividades de un proyecto representándose mediante las redes PERT-CPM.
3. Construir un proyecto de optimización, mediante la aplicación de las redes PERT-CPM.

Conducta de Entrada:

1. Al iniciar el uso de esta herramienta educativa, el aprendiz debe estar capacitado para manejar las operaciones básicas del computador.
2. Conocimientos matemáticos necesarios para poder abordar con éxito los modelos de redes.

Objetivos Específicos:

1. Reseñar las diversas terminologías empleadas en la construcción de redes.
2. Mencionar las reglas para la elaboración de una red.
3. Recordar la formulación de un programa de actividades de proyectos.
4. Observar los ejemplos demostrativos de la aplicación de una red.

Lección 2: Método de Revisión Hacia adelante.

Objetivos Terminales:

1. Reconocer los diferentes métodos y modelos asociados a las redes de la optimización.
2. Identificar un método que represente la solución a problemas reales enfrentados en la práctica profesional.

Conducta de Entrada:

1. Nociones básicas para el manejo de una computadora.
2. Conocimientos matemáticos necesarios para poder abordar con éxito esta lección.
3. Comprensión de la lección “Representación de la Red”.

Objetivos Específicos:

1. Citar al método de revisión hacia adelante como uno de los pasos para el cálculo de la ruta crítica.
2. Recordar las terminologías relacionadas con el método de revisión hacia adelante.
3. Identificar las diversas ecuaciones y variables relacionadas empleadas en el método de revisión hacia adelante.
4. Observar ejemplos demostrativos del método de revisión hacia adelante.

Lección 3: Método de Revisión Hacia Atrás.**Objetivos Terminales:**

1. Relacionar el método de revisión hacia atrás con las redes de la optimización.

2. Asociar el modelo de revisión hacia atrás a problemas existentes.

Conducta de Entrada:

1. Conocimientos para manejar las operaciones básicas del computador.
2. Nociones matemáticas necesarias para poder abordar con éxito los modelos de redes.
3. Recordar la lección “Representación de la Red”.

Objetivos Específicos:

1. Reconocer al método de revisión hacia atrás como uno de los pasos para el cálculo de la ruta crítica.
2. Nombrar las diversas terminologías, ecuaciones y variables relacionadas con el método de revisión hacia atrás.
3. Observar ejemplos demostrativos del método de revisión hacia atrás.

Lección 4: Redes con Incertidumbre:

Objetivos Terminales:

1. Entender las redes PERT-CPM con incertidumbre.
2. Comprender problemas de optimización a través de redes con incertidumbre.
3. Proponer la aplicación de redes con incertidumbre en proyectos de optimización.

Conducta de Entrada:

1. Conocer las operaciones básicas del computador.

2. Nociones matemáticas necesarias para poder abordar con éxito los modelos de redes.
3. Conocimientos básicos de estadísticas.
4. Comprender de la lección “Representación de la Red.”

Objetivos Específicos:

1. Recordar los casos de estimaciones en que PERT basa los tiempos de duración de una actividad.
2. Retener la formula para calcular un tiempo esperado de duración de una actividad.
3. Reconocer la varianza de los tiempos de actividad como parte de la solución a las redes con incertidumbre.
4. Observar el ejemplo demostrativo de redes con incertidumbre.

Lección 5: Análisis de redes considerando costos.

Objetivos Terminales:

1. Proponer el uso del análisis de redes considerando costos Construir a problemas administrativos.
2. Comprender los costos asociados a un proyecto.

Conducta de Entrada:

1. Al iniciar el uso de esta herramienta educativa, el aprendiz estará capacitado para manejar las operaciones básicas del computador.
2. Conocimientos matemáticos necesarios para poder abordar con éxito los modelos de redes.

Objetivos Específicos:

1. Mencionar el problema del flujo máximo como uno de los modelos matemáticos de redes.
2. Observar el ejemplo demostrativo de una red Pert-CPM considerando costos.

A continuación, se expone el contenido instruccional de la asignatura Modelos de Operaciones I.

Contenido Instruccional

Mediante el Contenido Instruccional, se establece un análisis específico de la estructuración del contenido de una lección de la aplicación. El esbozo de una de las lecciones que componen el software educativo permitirá conocer más detalladamente la funcionalidad y organización del contenido presentado en el resto de las lecciones y unidades.

En esta sección se realiza el análisis de la distribución del contenido en la ***Lección 1: Representación de la Red***, correspondiente a la ***Unidad 1: Análisis de Redes Pert-CPM***, en el SEMOI. A continuación, se especifican los ítems de esta lección, se seleccionará uno de estos ítems para su estudio y se mostraran las imágenes, actividad didáctica y la evaluación correspondiente a la unidad I.

1. Ítems

Mediante la especificación de ítems en cada lección, se establece y mantiene un orden en la forma como se muestra la información al usuario, facilitando la comprensión del material de estudio. Dentro del contenido del ***SEMOI*** se definirán los siguientes ítems para la ***Lección 1***, que hace referencia al tema de estudio ***“Representación de la Red”***:

1. Introducción.
2. Reglas para Construir una Red Pert-CPM.
3. Situaciones a evitar durante la construcción de redes.
4. Procedimiento para trazar la red medida.
5. Ejemplo de construcción de una red.

En la siguiente sección, se especificará uno de estos ítems, lo que permitirá, a su vez, exponer la manera como se mostrará el contenido estructural de cada tema de la lección.

2. Selección de Ítems

Dentro del ítem número 2 “Reglas para construir una red”, se especifican un conjunto de reglas que explican detalladamente cómo construir una red Pert-CPM.

En el texto que se presenta a continuación (Ver Fig. 4.7) se enumeran todas las normas que, según la información propuesta en el SEMOI, se deben seguir al momento de representar una red.

Dado un conjunto de actividades y sus relaciones de predecisión, se puede construir una representación grafica de acuerdo a lo siguiente:

1. El nodo 1 representa el inicio del proyecto. Por lo tanto, las actividades que parten del nodo 1 no pueden tener predecesoras.
2. El nodo Terminal o final del proyecto debe representar el término de todas las actividades incluidas en la red. Una actividad no puede ser representada por más de un arco en la red.
3. Dos nodos deben estar conectados por un solo arco. (No puede haber más de una flecha entre el mismo par de nodos). Ver figura 4.

4. Para no violar las reglas 3 y 4, a veces es necesario introducir una actividad **artificial (ficticia)** que posee tiempo de duración nulo. Por ejemplo, supongamos que las actividades A y B son predecesoras de la actividad C y además comienzan al mismo tiempo. Sin embargo, la red de la figura 3 viola la regla 4. Para corregir este problema, se introduce una actividad artificial indicada con un arco segmentado. En los casos en que haya necesidad de indicar que una actividad tiene una interrelación o continuación con otra se dibujará entre ambas una línea punteada, llamada liga, que tiene una duración de cero. La liga puede representar en algunas ocasiones un tiempo de espera para poder iniciar la actividad siguiente.
5. El evento final de una actividad será el evento inicial de la actividad siguiente.
6. Cada flecha (actividad) debe comenzar y terminar en un evento.
7. Las flechas indican solo precedencia lógica, ni su longitud ni su dirección tienen mayor significado. Las flechas no son vectores,

Fig. 4.7. Reglas para construir una red según contenido del SEMOI (1/2)

escalares ni representan medida alguna. No interesa la forma de las flechas, ya que se dibujarán de acuerdo con las necesidades y comodidad de presentación de la red. Pueden ser horizontales, verticales, ascendentes, descendentes curvas, rectas, quebradas, etc.

8. Todas las flechas de la red deben estar dirigidas más o menos de izquierda a derecha para mantener un orden lógico.

Fig. 4.7. Reglas para construir una red según contenido del SEMOI (2/2)

3. Imágenes

En la Lección 1 del “Análisis de Redes Pert-CPM”, además de mostrar el contenido referente a la construcción y representación de redes, se anexan diversas imágenes que apoyaran el proceso de comprensión y aprendizaje de este tema. A continuación, se muestran estas imágenes.

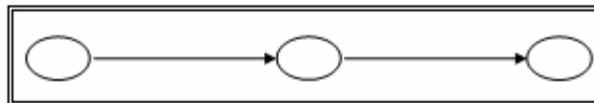


Fig. 4.8. Imagen 1.

La actividad A es predecesora de la actividad B

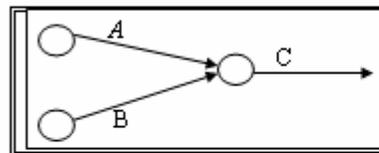


Fig. 4.9. Imagen 2.

Las actividades A y B son predecesoras de la actividad C.

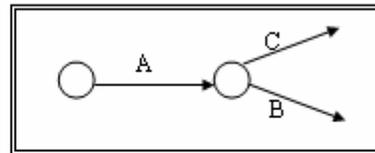


Fig. 4.10. Imagen 3.

La actividad A es predecesora de las actividades C y B.

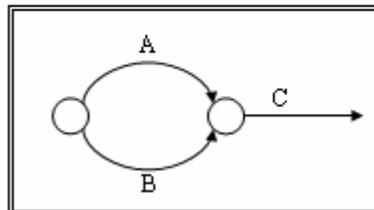


Fig. 4.11. Imagen 4

No puede haber más de una flecha entre el mismo par de nodos.

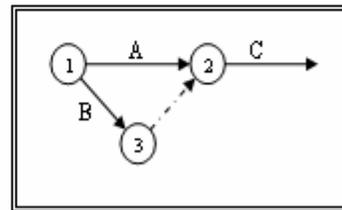


Fig. 4.12. Imagen 5.

Actividad ficticia entre los nodos 2 y 3.

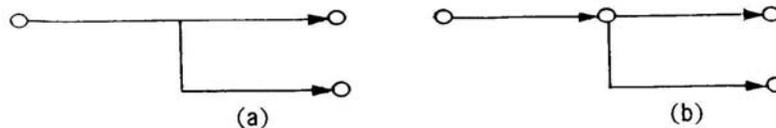


Fig. 4.13. Imagen 6.

Es incorrecto partir una actividad de una parte intermedia de otra actividad (a).
 Toda actividad debe empezar invariablemente en un evento y terminar en otro (b)

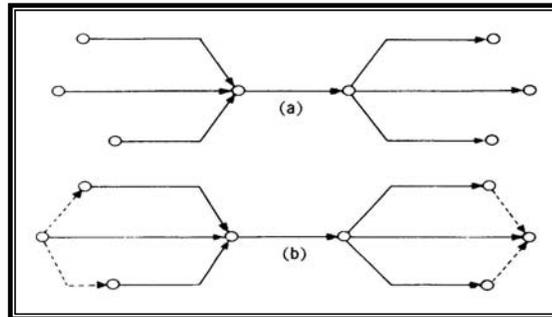


Fig. 4.14. Imagen 7.

Es incorrecto dejar nodos sueltos al terminar la red (a).
 Todos deben relacionarse con el evento inicial o final (b)

4. Actividad Didáctica

La actividad didáctica constituye un elemento clave para conducir el proceso didáctico y, por tanto, para la función educadora. El proceso de diseño de la actividad didáctica comienza con la selección del contenido para luego desarrollar la estructura de la actividad didáctica. Para ello, se toman en cuenta los siguientes criterios establecidos por Álvaro Galvis:

- Ofrecer indicaciones que ayuden al alumno a alcanzar la meta propuesta
- Permitir el control interactivo del material, a través de la computadora
- Combinar varios tipos de datos: texto, hipertexto, imágenes fijas e imágenes en movimiento.

Un bosquejo de la interfaz de la actividad didáctica de la Unidad 3: “Programación Dinámica” se muestra en la Figura 4.15, en la misma se observan las diferentes zonas establecidas para la pantalla de esta actividad:

- Zona 1: En esta zona se identifican los siguientes componentes: en la parte superior izquierda, el logo que identifica al software, en la parte superior centro, se observa la unidad a la cual pertenece la actividad didáctica
- Zona 2: En esta zona, se observa una breve descripción de la actividad a realizar.
- Zona 3: En esta zona se presenta la imagen correspondiente a la actividad didáctica que en este caso se trata de un rompecabezas. Esta es la zona activa en la cual el usuario podrá interactuar con la aplicación (armando el rompecabezas)
- Zona 4: Es la zona de control de la actividad didáctica. En esta, se muestra la tecla Mezclar, que permitirá desarmar el rompecabezas e iniciar el juego. Durante el juego, el cuadro de texto que señala ‘Felicidades’ marcará el número de piezas faltantes del rompecabezas.



Fig. 4.15. Identificación de las zonas en la pantalla de la Actividad Didáctica.

En cada unidad del SEMOI se realizará una actividad didáctica que complementará el proceso enseñanza-aprendizaje del tema. Solo se muestra como referencia la actividad didáctica de la unidad I, que, como se ha observado, se trata de un rompecabezas.

La imagen del rompecabezas corresponderá a una red Pert-CPM, y a medida que se arme el rompecabezas se simulará la construcción de la red.

El usuario deberá presionar el botón Mezclar para desorganizar las piezas del rompecabezas, a partir de ese momento podrá comenzar el juego. Las piezas se ubicarán de forma desordenada en el cuadro blanco ubicado en la parte derecha de la zona 3 de la Fig. 4.15. Desde el mismo, se irá seleccionando pieza por pieza y se deberán ubicar de manera correcta en la parte izquierda (como se muestra al principio del juego). En el cuadro de texto ubicado a la derecha del botón Mezclar se indicarán las piezas faltantes o mal ubicadas en el rompecabezas. Al ubicar correctamente todas

las piezas se termina el juego, el cuadro de texto deberá mostrara al usuario el mensaje 'FELICIDADES'.

5. Evaluación

Las evaluaciones se realizan por unidad. Cada evaluación consta de 10 preguntas con 4 opciones de respuesta cada una, el usuario deberá escoger la opción que considere correcta. Para la Unidad I, a la cual, corresponde la lección "Representación de la Red", el usuario deberá responder cada una de las preguntas formuladas. La evaluación de la unidad 1 consta igualmente de 10 preguntas, en las cuales, se evaluará la capacidad del estudiante para construir y analizar redes. Con 7 o más respuestas correctas se aprueba la evaluación. Cada pregunta tendrá un valor de 10 puntos. Por lo tanto, el mínimo aprobatorio será de 70 puntos.

Para la formulación de las preguntas se tiene en cuenta lo siguiente:

- Cada pregunta requiere solamente una respuesta.
- Las preguntas estarán adaptadas al nivel de habilidad y experiencia del estudiante con respecto a la unidad 1.
- Las preguntas se referirán a un sólo aspecto del análisis de redes Pert-CPM.
- La pregunta contiene frases que son idénticas a la información presentada en el contenido de la unidad 1.
- Las preguntas se referirán a los objetivos específicos de la unidad.
- Las preguntas son breves y claras.

En la Fig. 4.23 se muestra un bosquejo (diseño previo) de la interfaz de la evaluación de la Unidad 1 del SEMOI. La pantalla se divide en cuatro zonas de relevancia.

- **Zona 1:** Se ubica en la parte superior de la pantalla y permite la identificación de la unidad a la cual corresponde la evaluación.
- **Zona 2:** Esta zona se ubica en la parte izquierda de la pantalla. Se presentan las 10 preguntas de la evaluación con botones, esto permitirá al usuario, elegir la pregunta a responder en un momento determinado y en el orden que desee.
- **Zona 3:** Es la zona de contexto de la evaluación, donde se presentarán las preguntas y las opciones de respuestas. Se ubica en la parte derecha de la pantalla.
- **Zona 4:** Esta zona se ubica en la parte inferior de la pantalla y en ella se encuentra del lado izquierdo, el botón *FINALIZAR*, enlace que permite al estudiante obtener la calificación de la evaluación una vez haya respondido todas las preguntas. Esta zona, al igual que la zona 2, también permite controlar la pregunta donde esta ubicado el estudiante, ya que, proporciona la posibilidad de retroceder o avanzar a voluntad en una pregunta través de los botones *ATRÁS* y *ADELANTE* respectivamente ubicados en la parte derecha de esta zona.

Cada pregunta de la evaluación se presentará en una pantalla individual, donde cada usuario tendrá la posibilidad de elegir el orden en que desea contestarlas. Las preguntas podrán ser respondidas sin ningún orden específico realizando un click sobre el check de cada respuesta que considere correcta. El usuario podrá adelantar y retroceder cuantas veces lo desee.

Una vez hayan sido contestadas la totalidad de las respuestas, se procede a calificar la evaluación haciendo click sobre el botón Finalizar o en el botón Siguiente de la pregunta 10. El estudiante deberá responder todas las preguntas de la evaluación. En caso de que no responda todas las preguntas correspondientes, se mostrara un mensaje de advertencia como el de la Figura 4.24.

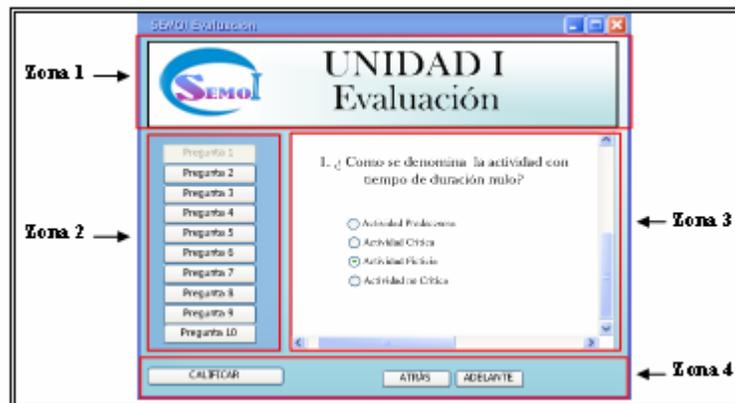


Fig. 4.23. Pantalla de la evaluación.

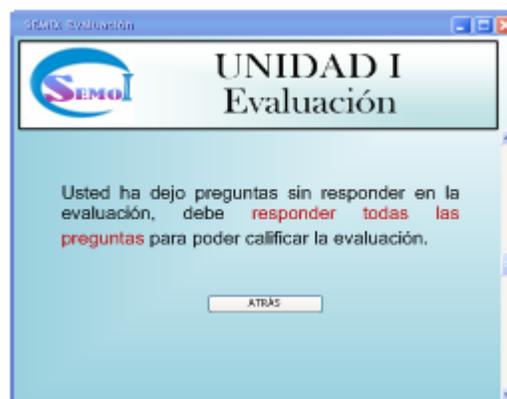


Fig. 4.24. Pantalla de advertencia de la evaluación.

La Figura 4.25. Presenta la pantalla de confirmación de calificación de la evaluación. Si el usuario está seguro de calificar la evaluación deberá hacer click en el botón CALIFICAR, de lo contrario deberá volver ATRÁS.



Fig. 4.25. Confirmación de calificación de evaluación.

En la Figura 4.26 se muestra la pantalla que revela la calificación obtenida. En la parte izquierda se presenta el conjunto de botones correspondientes a cada pregunta, identificando las preguntas que fueron contestadas de forma correcta o incorrecta. Estos botones permitirán al usuario visualizar la respuesta seleccionada por ellos para cada pregunta de su evaluación.

Finalmente, se tiene la pantalla de la Figura 4.27 en la cual se observa la retroalimentación de la pregunta 1 de la evaluación. En la parte inferior correspondiente a la zona de navegación se permitirá adelantar o retroceder preguntas para visualizar su respuesta. Al igual que en la pantalla anterior, en la parte izquierda de esta pantalla, se tienen botones que identifican cada una de las preguntas y permitirán al usuario seleccionar la que desee.



Fig. 4.26. Pantalla de evaluación mostrando la calificación.

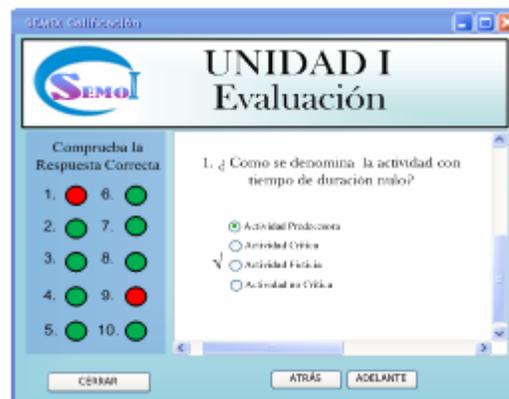


Fig. 4.27. Pantalla de evaluación mostrando la respuesta del usuario.

¿En que ambiente o micromundo aprender?: Micromundos interactivos

Es necesaria la formulación y creación de ambientes de trabajo o micromundos, en los que se establezcan situaciones interesantes, entretenidas y atractivas para lo que se desea enseñar. Micromundos que proporcionen y afiancen el conocimiento y además sirvan como marco de referencia para la interacción entre el usuario y el sistema, siendo un medio de gran relevancia para favorecer la interactividad y la participación del usuario.

Algunos de los micromundos interactivos propuestos como parte del diseño educativo, son:

- Micromundo “Armar una red”

Este micromundo (explicado en el apartado anterior) esta representado por un rompecabezas, el cual deberá ser armado correctamente por el usuario para completar la actividad. Inicialmente, el usuario tendrá una vista completa de la Figura (Red), luego deberá presionar el botón Mezclar para desordenar las piezas de la red. A partir de ese momento, podrá comenzar a armarla. Existirá un contador de las piezas faltantes, el cual, disminuirá a medida que se vaya colocando las piezas correctamente. Cuando se hayan colocado todas las piezas de manera correcta habrá concluido el juego.

- Micromundo “Formulación de juegos de 2 personas”

Se presentarán al usuario las estrategias de 2 empresas A y B de productos de consumo masivo que están estudiando las posibilidades de expandir su gama de productos en artículos de oficina, de limpieza y cosméticos. Los porcentajes del mercado capturado o perdido por la empresa A se representaran en una matriz. El usuario deberá determinar la solución óptima del juego y las estrategias puras asociadas para cada empresa utilizando los criterios mínimax y maximin.

- Micromundo “Crucigrama de LaGrange”

En este micromundo el usuario podrá evaluar los conceptos estudiados acerca del método de Lagrange para la resolución de problemas de programación no lineal. Para resolver el crucigrama, se dará un conjunto de pistas que permitirán completarlo. En caso de que el usuario acierte todos los términos ubicados en las líneas horizontales o verticales del mismo, se habrá concluido el juego, desabra perdido y tendrá la oportunidad de repetirlo, si lo desea.

¿Cómo motivar y mantener motivados a los usuarios?: Sistemas de Motivación.

Se debe crear un sistema de motivación que refuerce y promueva el aprendizaje de los estudiantes. El contenido software será presentado en un ambiente amigable exponiendo en cada una de las unidades imágenes y animaciones que llamen la atención del usuario y ayuden a mantenerlo la motivación por aprender.

Además se plantean los micromundos interactivos presentados en diversos modelos de juegos los cuales constituyen ambientes amigables e interesantes que despiertan la curiosidad y fomentan el aprendizaje.

Cada uno de estos sistemas de motivación se encuentran inmersos en diversas unidades, para ello se realizó un estudio previo para determinar las dificultades de cada tema y así asignar el elemento de motivación correcto que permita que el estudiante entienda y ejercite de forma más fácil el contenido de la asignatura.

¿Cómo saber que el aprendizaje se está logrando?: Evaluaciones

Para conocer el aprendizaje que se esta logrando, al finalizar cada unidad se plantea una situación de evaluación, que consiste en definir preguntas y problemas que sirvan como base para comprobar el conocimiento adquirido por el usuario. Las evaluaciones son elaboradas y redactadas en base a los objetivos planteados para cada una de las unidades, tomando siempre en cuenta que lo que se evalúa tenga estrecha relación con lo que se está enseñando.

Como fue mencionado anteriormente, la evaluación de cada una de las unidades estará conformada por 10 preguntas escogidas de forma aleatoria. Para cada pregunta se presentan las posibles opciones de respuestas de las cuales el usuario deberá seleccionar una (la que considere correcta). Al responder todas las preguntas

se debe presionar el botón Calificar, para conocer la calificación obtenida. El usuario tendrá la posibilidad de verificar cada una de las respuestas mediante la retroalimentación, en esta, se indicará si las respuestas seleccionadas son correctas o incorrectas. La evaluación se considerará aprobada si el estudiante responde correctamente 7 o más de las preguntas propuestas.

4.4.1.2 Diseño Comunicacional

En el diseño comunicacional se define la interfaz (zona de comunicación usuario-programa) de la aplicación, donde se diseñan los micromundos según las necesidades de la población objetivo y de acuerdo al contenido de la asignatura.

La interfaz es la zona de comunicación que permite la interacción entre el usuario y la aplicación. Ésta abarca todos los mensajes existentes entre el usuario y el programa y todos aquellos dispositivos de entrada y salida que permiten el intercambio de estos mensajes.

Para el SEMOI se utilizarán el teclado y el ratón como dispositivos físicos de entrada de información, las cornetas como dispositivos de salida y el monitor como dispositivo de entrada y salida, ya que, permite a la representación visual de la información de la aplicación, y a su vez, muestra las respuestas del usuario al interactuar con el software. Los mensajes se transmitirán básicamente mediante el monitor del computador teniendo especial cuidado con la distribución en pantalla de la información, el juego de colores, la forma y disposición de la información, la cantidad de ítems por pantalla, etc.

Se establecerán los botones e indicadores de enlaces como los mecanismos principales para la comunicación entre el usuario y la aplicación. Estos

proporcionarán al usuario, el acceso al contenido de cada unidad, evaluación, actividad didáctica, etc.

Definición de las Zonas de Comunicación entre Usuario y Programa

La distribución del contenido a tratar en el MEC, sugiere una selección de dispositivos para una interacción efectiva con los elementos del entorno. En la Figura 4.28 se muestra la organización de las zonas de comunicación entre el usuario y el programa.

La zona de comunicación, donde se combinan los dispositivos de entrada y salida, está compuesta por:

- Zona de trabajo, zona donde el usuario tiene a disposición la información proporcionada por el software: teorías, ejemplos y ejercicios; y en ellas se llevan a cabo las operaciones de recorrido que se desean efectuar sobre el objeto de estudio. Esta zona se ubicara en la parte central de la pantalla.
- Zona de control del programa, El control del programa tiene que ver con la posibilidad que debe tener el usuario de decidir cuando sigue la acción, a través del uso de botones, iconos y menús para seleccionar los temas. En éstas es posible alterar el flujo y el ritmo de ejecución del software. El control de flujo de ejecución suele estar asociado con la posible activación de los componentes del software educativo a partir de los menús de trabajo, esta zona también se relaciona con las posibilidades de abandono y reinicio del programa. En este caso, para la zona de control se seleccionan dos áreas: en una de ellas se ubicarán los Botones que darán acceso al usuario a las distintas unidades que conforma el software, estos están ubicados en la parte superior

de la pantalla, entre la Identificación del Software y la Identificación de la Unidad, por otra parte, se tiene el conjunto de botones que proporcionan acceso a diferentes zonas del contenido de la aplicación, se identifica como Zona de Navegación y esta ubicada en la parte inferior de la pantalla.

- Zona de contexto para la acción, a través de éstas el usuario puede saber en que unidad o modulo del sistema se encuentra, a que accesorios puede recurrir, como navegar por la aplicación, como escoger una opción, dar una respuesta. En esta zona se especifica el nombre del sistema y el nombre de la unidad que se está consultando. La zona de contexto se ubicará en 2 áreas en la parte superior de la pantalla: en la parte superior de la pantalla se ubica la Identificación del Software, y justo debajo de los botones superiores, se ubica la Identificación de la Unidad estudiada.



Fig. 4.28. Zonas de comunicación de la interfaz de usuario

Factores que Inciden en el Diseño de la Interfaz

Para comprobar que la interfaz satisface los requisitos mínimos, se toman en cuenta los factores como la ergonomía a nivel de hardware, software y comunicación.

- **Ergonomía a nivel de equipo.** A nivel de equipo se aplica a los dispositivos de E/S y al diseño de sus formas de uso, de manera que el abastecimiento y la obtención de información del computador se vea facilitada por tales mecanismos. Por ello, se debe considerar las alternativas que permitan atender las características de la población objetivo y la formas de adecuar los dispositivos de E/S a éstas, para poner a su disposición, medios de comunicación fáciles y prácticos de usar
- **Ergonomía a nivel de programación.** Es empleado a la especificación de los elementos del procesador dinámico y de las tareas de soporte. Las actividades de los procesadores de entrada y salida que maneja el procesador dinámico se especifican mediante la descripción funcional del sistema, es decir, de las funciones que éste realiza para cada tipo de usuario.
- **Ergonomía a nivel de comunicación.** Hacer el mejor uso de los equipos y de la programación para favorecer la efectividad en este proceso y aplicarlos al diseño de ambientes de intercomunicación. Algunas cualidades de comunicación importantes en las interfaces, son: consistencia a todo lo largo de la aplicación, simplicidad y adecuación a los usuarios.

Descripción de la Interfaz de Usuario

Se realiza el diseño de las distintas interfaces asentándose en diversas ideas que permitan al usuario navegar libremente a través de las unidades que componen al software. La información contenida en las pantallas se muestra de manera tal que el usuario siempre tenga el control sobre como dirigirse y que evento activar, logrando familiarizarse con las herramientas tales como: botones e indicadores de enlace.

Basándose en la interfaz mostrada por las diversas zonas de comunicación se insertan los elementos que le dan funcionalidad al software. En la Figura 4.29 se

observa la inserción de elementos que simbolizan métodos o atributos de las clases de diseño, o bien clases completas. Aquí se presenta una página que contiene un video de presentación o película de flash y una colección de anclas; estas son enlaces a páginas que corresponden al contenido de las unidades.

El ambiente de la aplicación se encuentra insertado a un objeto de tipo ventana (<<Window>>), el cual, contiene todas las instancias de las clases que intervienen en cada ejecución del programa. Este objeto se encuentra relacionado al sistema operativo del usuario.

Los botones ubicados en la parte superior constituyen el menú que permite el acceso a las diversas lecciones según sea la unidad a la que corresponde el botón y los botones que se encuentran en la parte inferior están diseñados para permitir el acceso a las funciones complementarias del software. Estos conjuntos de botones están conformados por objetos pertenecientes a la clase <<Button>>. Los objetos de las clases <<textos>> y <<Video>> van cambiando de acuerdo a la acción que se realice a través de los diversos botones.

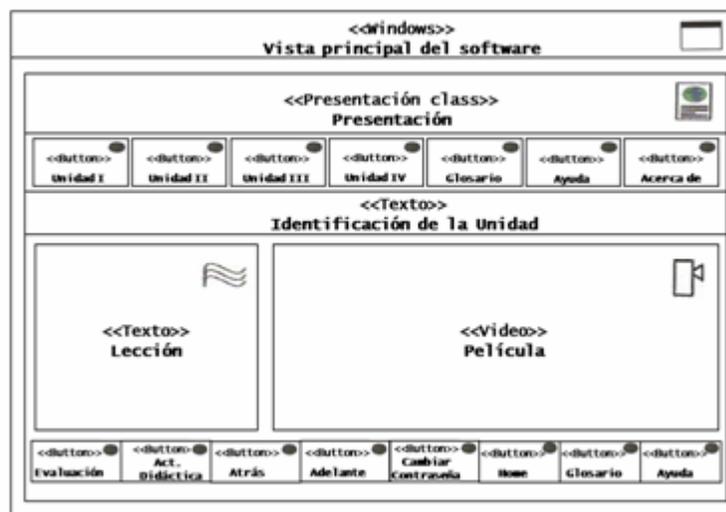


Fig. 4.29. Diagrama de la vista del contenido de las unidades.

4.4.1.3 Diseño Computacional

El diseño computacional define la estructura del software. Esta proporcionará un conjunto de modelos y diagramas que permitan representar en profundidad los aspectos relacionados con los requisitos funcionales, no funcionales y riesgos planteados con los componentes del sistema, esto permitirá modelar el sistema de manera que se satisfagan las necesidades y se mitiguen los riesgos. Además, permitirá la división de los trabajos de implementación en partes más manejables y más fáciles de desarrollar. La realización del diseño de la estructura del software contribuirá al establecimiento de una arquitectura estable y sólida y a crear un plano del modelo de implementación.

Modelo de Diseño

El modelo de diseño describe la realización física de los casos de uso centrándose en como los requisitos junto con las restricciones tienen impacto en el sistema a considerar y sirve como una entrada fundamental de las actividades de implementación. El modelo de diseño se representa por un sistema de diseño que denota el subsistema de nivel más alto del modelo. La utilización de otro subsistema es, entonces una forma de organización del modelo de diseño en porciones más manejables. [11]

- **Clases de Diseño** : Una clase de diseño es una abstracción “sin costuras” de una clase o construcción similar en la implementación del sistema. [11] Permite representar las clases del sistema, sus atributos, operaciones y asociaciones. Este diagrama se representa a través de un modelo Web que permite identificar las clases según el diseño (orientado a la Web) de la aplicación. WAE (Web Application Extension) es la notación utilizada

para la ejecución de este diagrama. Esta permite representar los componentes principales de diseño para un entorno Web. [Ver Cap II. 2.2.8.2 UML para la web]

En la Figura 4.30, se muestra un diagrama que expone la interacción entre las clases del caso de uso “Navegar Contenido de Unidades”. Este diagrama esta compuesto por un conjunto de clases que representan la interacción del usuario con el sistema. La clase Principal Unidades permite el acceso a la totalidad de funciones que corresponden al caso de uso navegar contenido de unidades. Este incluye estudio de las lecciones, la consulta de glosario y ayuda, la realización de la actividad didáctica de la unidad, la ejecución de la evaluación y el cambio de contraseña del usuario. También se observa que desde la interfaz de lección se puede acceder a otra unidad, consultar el glosario y la ayuda y/o solicitar el cambio de la contraseña. Para ello, se requiere del establecimiento de la Conexión, las consultas a la tabla de Usuarios para el cambio de contraseña y las consultas a la tabla pregunta y a la tabla de respuestas de la base de datos, para la evaluación. Desde la página Principal Unidades se solicitan las unidades almacenadas en la carpeta denominada con el mismo nombre, como se puede observar en el diagrama. Igualmente, cada lección y animación se obtendrán de las carpetas lecciones y animaciones, respectivamente.

En la Figura 4.31, por otra parte, se muestra el diagrama de clases de diseño Web para el caso de uso “Configurar Sistema”. En este diagrama se parte de la interfaz principal de Configuración del sistema que es la encargada de redireccionar al administrador a las funciones que le corresponden. A partir del <<form>> Configuración del Sistema, se proporcionan diferentes opciones al administrador, el cual puede escoger entre el Registro de un Usuario, la Actualización de Datos de algún

Tomando como base el diagrama de clases de diseño mostrado en la Fig. 4.30 se desarrolla el diagrama de secuencia del caso de uso “Navegar Contenido de Unidades” (Ver Fig. 4.32).

De acuerdo con este diagrama se tienen las siguientes secuencias:

1. Solicitar el cambio de contraseña de usuario.
2. Solicitar consulta de Glosario.
3. Solicitar consulta de Ayuda del sistema.
4. Realizar la evaluación correspondiente a la unidad de estudio.
5. Realizar la Actividad Didáctica de la unidad.

Cada una de estas actividades, podrán ser realizadas por usuarios y por administradores del sistema, quienes utilizarán el mismo flujo de acciones, para cada caso.

Por otra parte se desarrolla el diagrama de secuencia del caso de uso “Configurar Sistema” tomando como base el diagrama de clases de diseño de la Fig. 4.31 (Ver Figura 4.33)



Fig. 4.32. Diagrama de Secuencia de "Navegar Contenido de Unidades"

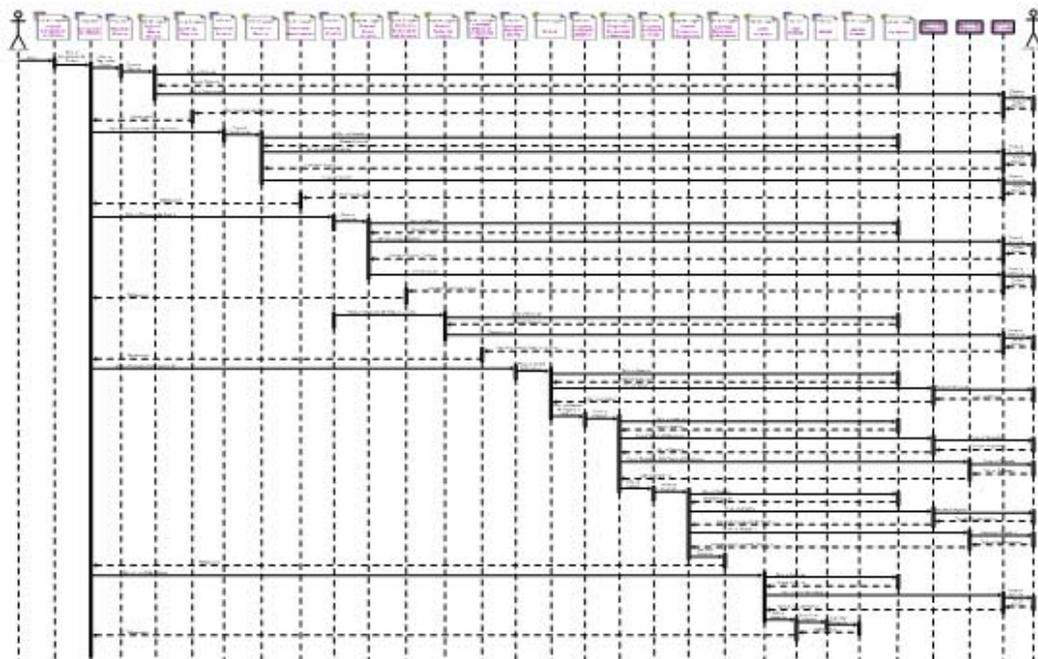


Fig. 4.33. Diagrama de Secuencia de "Configurar Sistema"

En este diagrama se observan las siguientes secuencias:

1. Registrar nuevo usuario.
2. Actualizar datos de usuarios registrados.
3. Eliminar algún usuario.
4. Eliminar Todos los alumnos (usuarios) registrados.
5. Configurar evaluaciones.
6. Listar Alumnos (usuarios) Registrados.
7. Imprimir alumnos (usuarios) Registrados.
8. Explorar unidades.

Estas actividades solo podrán ser realizadas por el administrador. Estas secuencias producirán cambios directamente en la base de datos del sistema.

- **Subsistema de Diseño:** Los subsistemas de diseño son unas formas de organizar los artefactos del modelo de diseño en piezas más manejables. Un subsistema puede constar de clases del diseño, realizaciones de casos de uso, interfaces y otros subsistemas.

Los subsistemas pueden identificarse inicialmente como forma de dividir el trabajo de diseño, o bien pueden irse encontrando a medida que el modelo de diseño evoluciona y va “creciendo” hasta convertirse en una gran estructura que debe ser descompuesta. [11]

Los subsistemas de diseño se representan mediante el diagrama de capas del sistema. Para la elaboración del diagrama de capas inicialmente, se deben identificar los subsistemas pertenecientes a las capas específica de la aplicación y general de la aplicación. Para ello, se

toma como base la descomposición en paquetes de análisis (realizada previamente) lo que permitirá identificar y, adicionalmente, refinar los correspondientes subsistemas dentro del modelo de diseño.

Para la capa específica de la aplicación se identifican los siguientes subsistemas: Gestión de Actividad Didáctica, Gestión de Asistencia al Usuario y Gestión de Interacción con Lección, mientras que para la capa general de la aplicación se distinguen los subsistemas: Gestión de Interacción con Contenido de Unidades, Gestión de Evaluación y se incluye además el subsistema Gestión de Identificación de Usuario, que se encarga de procesar el login de los usuarios.

Luego, se identifican los subsistemas de las capas intermedia y de software del sistema, que constituyen los cimientos de un sistema, ya que, toda la funcionalidad descansa sobre software como sistemas operativos, sistemas de gestión de base de datos, software de comunicaciones, entre otros. A la capa intermedia corresponden los subsistemas: Páginas Html, Servidor Html, Macromedia Flash, Reproductor Flash Player, Navegador Web y PHP. Para la capa de software se identifica el Protocolo TCP/IP y el manejador de Base de Datos MySQL.

Finalmente, se definen las dependencias entre los subsistemas si sus contenidos tienen relación unos con otros. La estructuración de todos los subsistemas en capas y sus dependencias (relaciones) se muestran a continuación en la Figura 4.34.

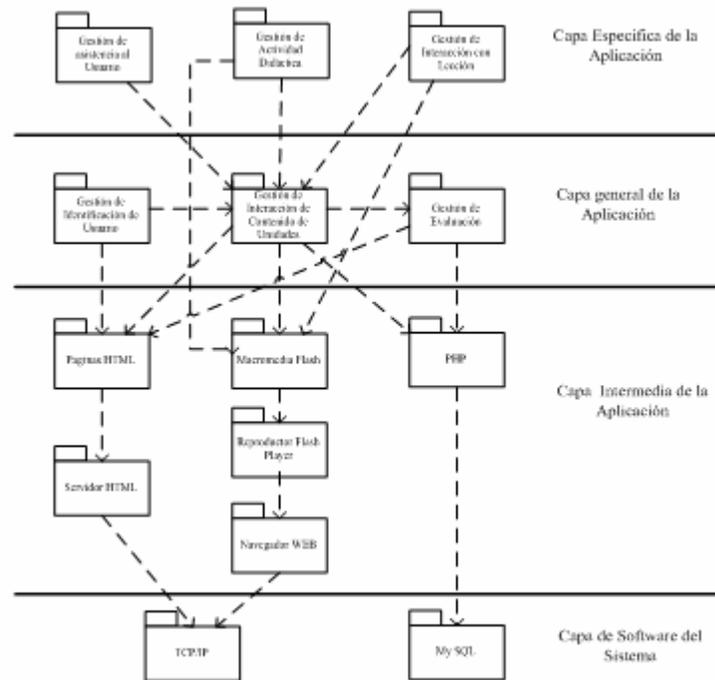


Fig. 4.34. Diagrama de capas del sistema.

• Modelo de Navegación

Este modelo proporciona la representación y documentación de la estructura navegable del sistema de forma sencilla y fácil de comprender, mostrando la forma como navegará cada usuario a través de la aplicación.

Para los usuarios del SEMOI existen 2 modalidades de navegación: la primera involucra al administrador del sistema. Es de carácter privado y representa las tareas propias del administrador en el sistema. Entre las tareas propias del administrador se pueden observar las conFIGuraciones

de usuario (registro, actualización, borrado y listado) y de las evaluaciones. En la Figura 4.35 se observa este modelo de navegación.

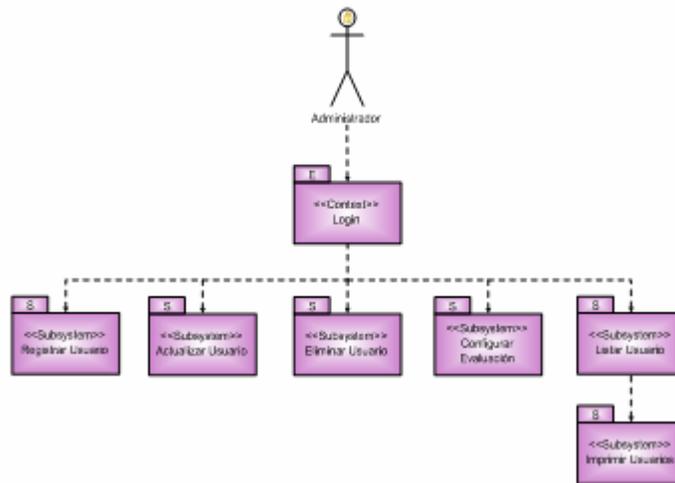


Fig. 4.35. Diagrama de navegación (Modalidad 1: Privado-Administrador)

Para el usuario existe el modelo de navegación igualmente privado debido a que requiere el login del alumno para permitir el acceso de las funcionalidades del software. El administrador también tendrá acceso a esta modalidad. Entre estas funcionalidades se menciona la navegación del contenido de las unidades, el cambio de contraseña, consultas al glosario y ayuda, y ejecución de actividades didácticas y evaluaciones. (Ver Fig. 4.36.)

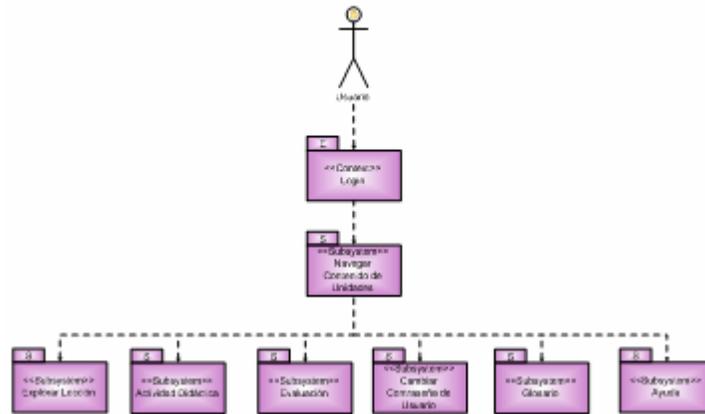


Fig. 4.36. Diagrama de navegación (Modalidad 2: Privado-Usuario)

4.4.2 Diseño de la Base de Datos

La base de datos del sistema es utilizada para almacenar las Preguntas y Respuestas de las evaluaciones de cada unidad de estudio del software. Para su representación, se utiliza el modelo Entidad-Relación, que permite modelar conceptualmente los datos del sistema a través del establecimiento y representación de las entidades sus relaciones, facilitando la creación de la Base de Datos.

Adicionalmente, la Base de Datos contará con la entidad Usuarios que permitirá almacenar las contraseñas de inicio de sesión para cada usuario del software. Se almacenarán, además, otros datos de relevancia del usuario como nombre, apellido, correo electrónico, entre otros.

El modelo Entidad-Relación consta de entidades y sus atributos, relaciones, roles y claves que permitirán la representación gráfica de la base de datos. En la Figura 4.37, se observa la representación de las entidades de la base de datos. Estas entidades representan las tablas del sistema, y son denominadas Preguntas,

Respuestas y Usuarios, cada una se muestra con sus respectivos atributos. Las entidades Preguntas y Respuestas, están interconectadas mediante la relación Poseen, debido a que cada respuesta corresponde a una pregunta específica. Para la identificación de la clave principal de cada tabla se subraya el atributo correspondiente a la misma.

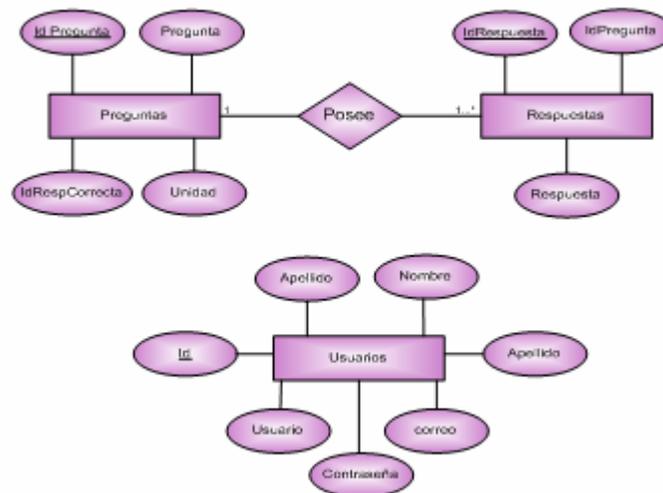


Fig. 4.37. Diagrama Entidad-Relación SEMOI.

En la tabla 4.3 Se encuentran especificados los atributos de la entidad Preguntas. Por su parte, en la tabla 4.4, se especifican los atributos de la entidad Respuestas. Y finalmente, en la tabla 4.5, se muestran los atributos de la entidad Usuarios.

Tabla. 4.3. Descripción de los atributos de la entidad Preguntas

ATRIBUTOS	DESCRIPCIÓN
IdPregunta	Almacena un número que identifica cada pregunta de la evaluación. Es la clave principal. Campo de Tipo entero de 3 dígitos.
Pregunta	Almacena cada pregunta de la evaluación. Campo de tipo Varhar de 500 caracteres.
IdRespCorrecta	Almacena un número que identifica la respuesta correcta de la pregunta a la cual corresponde. Campo de tipo entero de 4 dígitos.
Unidad	Almacena el número que identifica a la unidad a la cual pertenece esa evaluación. Campo de tipo entero de 1 dígito.

Tabla. 4.4. Descripción de los atributos de la entidad Respuestas

ATRIBUTOS	DESCRIPCIÓN
IdRespuesta	Almacena un número que identifica cada respuesta de la evaluación. Es la clave principal. Campo de Tipo entero de 4 dígitos.
Respuesta	Almacena la(s) respuesta(s) correctas de cada pregunta de la evaluación. Campo de tipo varchar de 500 caracteres.
IdPregunta	Clave Foránea, que relaciona a la tabla respuesta con la tabla pregunta, ya que, almacena el id de la pregunta a la cual corresponde. Campo de tipo varchar de 3 caracteres.

Tabla. 4.5. Descripción de los atributos de la entidad Usuarios

ATRIBUTOS	DESCRIPCIÓN
Id	Atributo que almacena el código correspondiente de identificación de cada usuario. Campo de tipo entero de 4 dígitos.

Usuarios	Identifica al usuario. Almacena el nombre de registro del usuario. Le permitirá logearse en el sistema. Campo de tipo varchar de 20 caracteres.
Contraseña	Almacena la contraseña correspondiente a cada usuario. Campo de tipo varchar de 8 caracteres.
Nivel	Campo que especifica los privilegios del usuario. Se emplea el valor de uno (1) para privilegios de administrador y dos (2) para privilegios de usuario. Campo de tipo entero de 1 dígito.
Nombre	Almacena el nombre del usuario. Campo de tipo varchar de 20 caracteres.
Apellido	Almacena el apellido del usuario. Campo de tipo varchar de 20 caracteres.
Correo	Almacena el correo electrónico del usuario. Campo de tipo varchar de 35 caracteres.

4.5 FLUJO DE TRABAJO: IMPLEMENTACIÓN

Este apartado corresponde al desarrollo del software educativo, ya que, se refiere a la implementación de la estructura de las interfaces (componentes), definidos en el diseño comunicacional y computacional. Para ello, se emplea Macromedia Dreamweaver 8 (editor de páginas PHP y HTML), Macromedia flash 8 (para la creación de animaciones) y Macromedia Fireworks 8 (para creación de imágenes).

Para la implementación se toman los resultados producidos en el flujo de trabajo de diseño y se implementan en términos de componentes, es decir, código fuente, script, ejecutables, entre otros.

La mayor parte de la arquitectura del sistema es capturada en el diseño, por lo tanto, al flujo de trabajo de implementación corresponde el desarrollo de la arquitectura y el sistema como un todo.

En este flujo se trabajará, específicamente, con el caso de uso Navegar Contenido de Unidades, con el objetivo de explicar la metodología empleada para su implementación, y mostrando sus componentes. Este caso de uso al igual que Configurar Sistema, se estudiarán en profundidad en la fase de Construcción.

4.5.1 Modelo de Implementación

El modelo de implementación describe como los elementos del modelo de diseño, como las clases se implementan en términos de componentes, describe también como se implementan estos componentes de acuerdo con los mecanismos de estructuración modularización disponibles en el entorno de implementación y en lenguaje de programación utilizado, y en como dependen los componentes unos de otros. [11]

Para su representación se utiliza el diagrama de componentes de la Figura 4.39. En este, se muestran, los componentes principales requeridos para la implementación del caso de uso “Navegar Contenido de Unidades”. Se indica el tipo de archivo correspondiente y su extensión (flash, html, php). La relación entre cada componente se establece a través de la notación <<trace>> y representa la traza que proporciona el vínculo entre los componentes que han sido desarrollados hasta ahora. Específicamente, se expone la iteración de la lección 1 correspondiente a la unidad I (Análisis de Redes Pert CPM) del software educativo.

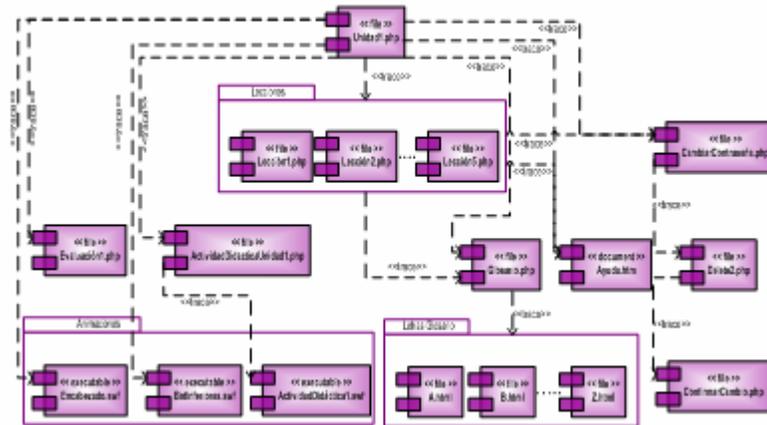


Fig. 4.39. Diagrama de Componentes Parcial del Sistema SEMOI.

4.5.2 Implementación de las Clases

El propósito de la implementación de una clase es implementar una clase de diseño en un componente fichero. Esto incluye lo siguiente:

- Esbozo de un componente fichero que contendrá el código fuente.
- Generación del código fuente a partir de la clase de diseño y de las relaciones en que participa.
- Implementación de las operaciones de la clase de diseño en forma de métodos.
- Comprobación de que el componente proporciona las mismas interfaces que la clase de diseño. [11]

Para esta fase se realiza la implementación de los componentes mostrados en el diagrama de la Figura 4.39.

Una vez, que se ha verificado la identidad del usuario, este podrá acceder a la clase unidades que permitirá navegar por las unidades y lecciones del software, donde se proporcionan links que permiten el acceso al glosario, ayuda, la actividad didáctica, y a la evaluación de la unidad, además proporciona la posibilidad de realizar el cambio de la contraseña de usuario, si así lo desea. (Ver Fig.4.40).

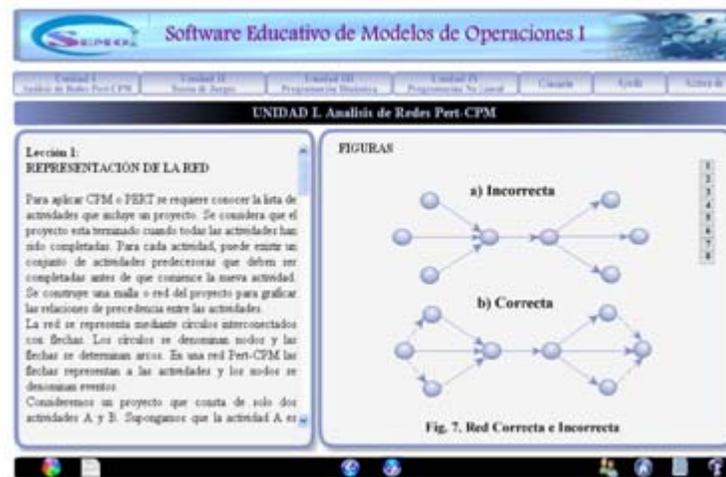


Fig.4.40. Interfaz Unidades

El código correspondiente a la interfaz anterior (unidadIleccionI.php) se muestra a continuación, y esta constituido principalmente por sentencias Html.

Se inicia con `session_start();`, la cual, es una sentencia php que permite el registro de sesiones de usuario y a su vez, permite mejorar el control de las operaciones de cada usuario. Posteriormente, continúa con la especificación de capas y estilos (css) de la página.

```
<?php session_start(); ?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>SEMOI_UNIDAD_I_ Lecci&oacute;n_1: Representaci&oacute;n de la Red</title>
<style type="text/css">
<!--
#Layer1 {
    position:absolute;
    left:1%;
    top:97px;
    width:100%;
    height:35px;
    z-index:1;
}
#Layer4 {
    position:absolute;
    left:25%;
    top:135px;
    width:50%;
    height:31px;
    z-index:1;
}
#Layer2 {
    position:absolute;
    left:25%;
    top:140px;
    width:50%;
    height:25px;
    z-index:1;
}
.Estilo11 {
    color: #B8C0DD;
    font-weight: bold;
    font-size: 16px;
}
.Estilo12 {font-size: 18px}
#Layer3 {
```

```
    position:absolute;
    left:5%;
    top:189px;
    width:37%;
    height:345px;
    z-index:1;
    overflow: auto;
}
#Layer5 {
    position:absolute;
    left:47%;
    top:207px;
    width:400px;
    height:338px;
    z-index:2;
    background-color: #3300FF;
}
#Layer6 {
    position:absolute;
    left:44%;
    top:206px;
    width:50%;
    height:338px;
    z-index:2;
}
#Layer7 {
    position:absolute;
    left:45%;
    top:184px;
    width:10%;
    height:27px;
    z-index:1;
}
-->
```

El siguiente código en JavaScript es el encargado de la creación y manejo del menú desplegable mostrado para cada una de las lecciones de las unidades del software. Para ello, se exportan las funciones requeridas del archivo mm_menu.js.

```

</style>
<script language="JavaScript">
<!--
function mmLoadMenus() {
    if (window.mm_menu_0318134401_0) return;
        window.mm_menu_0318134401_0 = new Menu("root",261,13,"Georgia, Times New
Roman, Times, serif",9,"#000000","#FFFFFF","#CCCCCC","#800080","left","middle",2,0,800,-
5,7,true,true,true,0,true,true);
        mm_menu_0318134401_0.addItem("Introducción","location='unidad1Introduccion.p
hp'");
        mm_menu_0318134401_0.addItem("Representacion&nbsp;de&nbsp;la&nbsp;Red", "
location='unidad1leccion1.php'");
        mm_menu_0318134401_0.addItem("Método&nbsp;de&nbsp;Revisión&nbsp;Hacia
&nbsp;Adelante","location='unidad1leccion2.php'");
        mm_menu_0318134401_0.addItem("Método&nbsp;de&nbsp;Revisión&nbsp;Hacia
&nbsp;Atrás","location='unidad1leccion3.php'");
        mm_menu_0318134401_0.addItem("Redes&nbsp;con&nbsp;Incertidumbre","locatio
n='unidad1leccion4.php'");
        mm_menu_0318134401_0.addItem("Análisis&nbsp;de&nbsp;Redes&nbsp;Consider
ando&nbsp;Costos","location='unidad1leccion5.php'");
        mm_menu_0318134401_0.hideOnMouseOut=true;
        mm_menu_0318134401_0.bgColor='#555555';
        mm_menu_0318134401_0.menuBorder=1;
        mm_menu_0318134401_0.menuLiteBgColor='#FFFFFF';
        mm_menu_0318134401_0.menuBorderBgColor='#0000FF';
        window.mm_menu_0318142419_0 = new Menu("root",301,13,"Georgia, Times New
Roman, Times, serif",9,"#000000","#FFFFFF","#CCCCCC","#800080","left","middle",2,0,800,-
5,7,true,true,true,0,true,true);

```

```

mm_menu_0318142419_0.addItem("Introducción","location='unidad2Introduccion.ph
p'");
mm_menu_0318142419_0.addItem("Formulación&nbsp;de&nbsp;Juegos&nbsp;de&
nbsp;2&nbsp;Personas","location='unidad2leccion1.php'");
mm_menu_0318142419_0.addItem("Formulacion&nbsp;de&nbsp;Juegos&nbsp;Con
&nbsp;Estrategias&nbsp;Mixtas","location='unidad2leccion2.php'");
mm_menu_0318142419_0.hideOnMouseOut=true;
mm_menu_0318142419_0.bgColor='#555555';
mm_menu_0318142419_0.menuBorder=1;
mm_menu_0318142419_0.menuLiteBgColor='#FFFFFF';
mm_menu_0318142419_0.menuBorderBgColor='#0000FF';
window.mm_menu_0318142839_0 = new Menu("root",279,13,"Georgia, Times New
Roman, Times, serif",9,"#000000","#FFFFFF","#CCCCCC","#800080","left","middle",2,0,800,-
5,7,true,true,true,0,true,true);
mm_menu_0318142839_0.addItem("Introducción","location='unidad3Introduccion.p
hp'");
mm_menu_0318142839_0.addItem("Recursión&nbsp;en&nbsp;Reversa&nbsp;y&nb
sp;en&nbsp;Avance","location='unidad3leccion1.php'");
mm_menu_0318142839_0.addItem("Modelo&nbsp;de&nbsp;la&nbsp;Ruta&nbsp;M
ás&nbsp;Corta","location='unidad3leccion2.php'");
mm_menu_0318142839_0.addItem("Modelos&nbsp;de&nbsp;Tamaño&nbsp;de&nb
sp;Fuerza&nbsp;de&nbsp;Trabajo","location='unidad3leccion3.php'");
mm_menu_0318142839_0.addItem("Aplicaciones&nbsp;de&nbsp;la&nbsp;Ppogram
ación&nbsp;Dinámica","location='unidad3leccion4.php'");
mm_menu_0318142839_0.hideOnMouseOut=true;
mm_menu_0318142839_0.bgColor='#555555';
mm_menu_0318142839_0.menuBorder=1;
mm_menu_0318142839_0.menuLiteBgColor='#FFFFFF';
mm_menu_0318142839_0.menuBorderBgColor='#0000FF';
window.mm_menu_0318143846_0 = new Menu("root",266,13,"Georgia, Times New
Roman, Times, serif",9,"#000000","#FFFFFF","#CCCCCC","#800080","left","middle",2,0,800,-
5,7,true,true,true,0,true,true);
mm_menu_0318143846_0.addItem("Introducción","location='unidad4Introduccion.p
hp'");

```

```

mm_menu_0318143846_0.addItem("Función&nbsp;Cónca&nbsp;y&nbsp;Convex
a","location='unidad4leccion1.php'");
mm_menu_0318143846_0.addItem("Optimización&nbsp;No&nbsp;Restringida","loc
ation='unidad4leccion2.php'");
mm_menu_0318143846_0.addItem("Optimización&nbsp;Restringida","location='uni
dad4leccion3.php'");
mm_menu_0318143846_0.addItem("Método&nbsp;de&nbsp;Multiplicadores&nbsp;
de&nbsp;LaGrange","location='unidad4leccion4.php'");
mm_menu_0318143846_0.hideOnMouseOut=true;
mm_menu_0318143846_0.bgColor='#555555';
mm_menu_0318143846_0.menuBorder=1;
mm_menu_0318143846_0.menuLiteBgColor='#FFFFFF';
mm_menu_0318143846_0.menuBorderBgColor='#0000FF';
mm_menu_0318143846_0.writeMenus();
} // mmLoadMenus()
function MM_showMenu(menu, x, y, child, imgname) {
if (!window.mmWroteMenu) return;
MM_clearTimeout();
if (menu) {
var obj = FIND(imgname) || document.images[imgname] || document.links
[imgname] || document.anchors[imgname];
x = moveXbySlicePos (x, obj);
y = moveYbySlicePos (y, obj);
}
if (document.layers) {
if (menu) {
var l = menu.menuLayer || menu;
l.top = l.left = 1;
hideActiveMenus();
if (this.visibility) l = this;
window.ActiveMenu = l;
} else {
var l = child;
}
}
}

```

```

if (!l) return;
for (var i=0; i<l.layers.length; i++) {
    if (!l.layers[i].isHilite) l.layers[i].visibility = "inherit";
    if (l.layers[i].document.layers.length > 0) MM_showMenu(null, "relative",
"relative", l.layers[i]);
}
if (l.parentLayer) {
    if (x != "relative") l.parentLayer.left = x || window.pageX || 0;
    if (l.parentLayer.left + l.clip.width > window.innerWidth) l.parentLayer.left
-= (l.parentLayer.left + l.clip.width - window.innerWidth);
    if (y != "relative") l.parentLayer.top = y || window.pageY || 0;
    if (l.parentLayer.isContainer) {
        l.Menu.xOffset = window.pageXOffset;
        l.Menu.yOffset = window.pageYOffset;
        l.parentLayer.clip.width = window.ActiveMenu.clip.width +2;
        l.parentLayer.clip.height = window.ActiveMenu.clip.height +2;
        if (l.parentLayer.menuContainerBgColor &&
l.Menu.menuBgOpaque ) l.parentLayer.document.bgColor = l.parentLayer.menuContainerBgColor;
    }
}
l.visibility = "inherit";
if (l.Menu) l.Menu.container.visibility = "inherit";
} else if (FIND("menuItem0")) {
    var l = menu.menuLayer || menu;
    hideActiveMenus();
    if (typeof(l) == "string") l = FIND(l);
    window.ActiveMenu = l;
    var s = l.style;
    s.visibility = "inherit";
    if (x != "relative") {
        s.pixelLeft = x || (window.pageX + document.body.scrollLeft) || 0;
        s.left = s.pixelLeft + 'px';
    }
    if (y != "relative") {

```

```

        s.pixelTop = y || (window.pageY + document.body.scrollTop) || 0;
        s.top = s.pixelTop + 'px';
    }
    l.Menu.xOffset = document.body.scrollLeft;
    l.Menu.yOffset = document.body.scrollTop;
}
if (menu) window.activeMenus[window.activeMenus.length] = l;
MM_clearTimeout();
}
//-->
</script>
<script language="JavaScript" src=" mm_menu.js"> </script>
</head>

```

En el cuerpo (body) del archivo se especifica el contenido de la Lección 1 (Representación de la Red) de la Unidad I - Análisis de Redes Pert-CPM. Se incluye el código que permite exportar las animaciones e imágenes de la lección.

```

<body>
<div class="Estilo11" id="Layer2">
<div align="center" class="Estilo12">UNIDAD I. Analisis de Redes Pert-CPM </div>
</div>
<div id="Layer3">
<p><strong>Lección 1: <br />
REPRESENTACIÓN DE LA RED</strong></p>
<p align="justify"><a name="_Toc106212748" id="_Toc106212748"></a>Para aplicar CPM
o PERT se requiere conocer la lista de actividades que incluye un proyecto. Se considera que el
proyecto está terminado cuando todas las actividades han sido completadas. Para cada actividad,
puede existir un conjunto de actividades predecesoras que deben ser completadas antes de que
comience la nueva actividad. Se construye una malla o red del proyecto para graficar las relaciones de
precedencia entre las actividades. <br />

```

La red se representa mediante círculos interconectados con flechas. Los círculos se denominan nodos y las flechas se determinan arcos. En una red Pert-CPM las flechas representan a las actividades y los nodos se denominan eventos.

Consideremos un proyecto que consta de solo dos actividades A y B. Supongamos que la actividad A es predecesora de la actividad B. La representación gráfica de este proyecto se muestra en la **figura 1**. Asimismo, el nodo 2 representa la culminación de la actividad A y el comienzo de la actividad B.

Si suponemos ahora que las actividades A y B deben ser terminadas antes que una actividad C pueda comenzar, la malla del proyecto queda como se muestra en la **figura 2**. En este caso, el nodo representa que las actividades A y B se han terminado, además del inicio de la actividad C. Si la actividad A fuera predecesora de las actividades B y C, la red quedara como se muestra en la **figura 3**.

Reglas Para Construir Una Red

Dado un conjunto de actividades y sus relaciones de precedencia, se puede construir una representación gráfica de acuerdo a lo siguiente:

1. El nodo 1 representa el inicio del proyecto. Por lo tanto, las actividades que parten del nodo 1 no pueden tener predecesoras.
2. El nodo Terminal o final del proyecto debe representar el término de todas las actividades incluidas en la red.
3. Una actividad no puede ser representada por más de un arco en la red.
4. Dos nodos deben estar conectados por un solo arco. (No puede haber más de una flecha entre el mismo par de nodos).

Para no violar las reglas 3 y 4, a veces es necesario introducir una actividad artificial (ficticia) que posee tiempo de duración nulo. Por ejemplo, supongamos que las actividades A y B son predecesoras de la actividad C y además comienzan al mismo tiempo. En este caso, una primera representación podría ser la indicada en la **figura 4**. Sin embargo, la red de la **figura 4** viola la regla 4. Para corregir este problema, se introduce una actividad artificial indicada con un arco segmentado en la **figura 5**.

En los casos en que haya necesidad de indicar que una actividad tiene una interrelación o continuación con otra se dibujan entre ambas una línea punteada, llamada *liga*, que tiene una duración de cero.

La liga puede representar en algunas ocasiones un tiempo de espera para poder iniciar la actividad siguiente.

- El evento final de una actividad será el evento inicial de la actividad siguiente.

- Cada flecha (actividad) debe comenzar y terminar en un evento.

- Las flechas indican solo precedencia lógica, ni su longitud ni su dirección tienen mayor significado. Las flechas no son vectores, escalares ni representan medida alguna. No interesa la forma de las flechas, ya que se dibujarán de acuerdo con las necesidades y comodidad de presentación de la red. Pueden ser horizontales, verticales, ascendentes, descendentes curvas, rectas, quebradas, etc.

- Todas las flechas de la red deben estar dirigidas más o menos de izquierda a derecha para mantener un orden lógico.

Al construir la red, debe evitarse lo siguiente:

- Dos actividades que parten de un mismo evento y llegan a un mismo evento. Esto produce confusión de tiempo y de continuidad. Debe abrirse el evento inicial o el evento final en dos eventos y unirlos con una liga.

- Partir una actividad de una parte intermedia de otra actividad. Toda actividad debe empezar invariablemente en un evento y terminar en otro. Cuando se presenta este caso, a la actividad base o inicial se le divide en eventos basándose en porcentajes y se derivan de ellos las actividades secundadas. Ver figura 6.

- Dejar eventos sueltos al terminar la red. Todos ellos deben relacionarse con el evento inicial o con el evento final. Ver figura 7.

Procedimiento Para Trazar la Red Medida

Para ello se denominará evento *i* al nodo inicial y evento *j* al nodo final.

Para dibujar la red, se inicia dibujando las actividades que parten del evento *i*. Cada una de ellas debe dibujarse de tal manera que el evento *j* termine de acuerdo con la duración establecida, en el tiempo indicado. Ahora mostraremos la iniciación de las actividades 1, 2, 3, y 4 con duración de tres, dos, tres y cinco días respectivamente.

Posteriormente, se continúa la construcción de la red de manera ordenada con base en las actividades que tienen como precedente inmediato aquellas actividades que ya han

sido representadas. La construcción de la red culmina una vez sea representado en nodo o evento final. Si es necesario se deben incluir actividades ficticias.

Se deben representar el código de la actividad y se debe identificar los nodos de manera que indiquen una secuencia lógica en la red.

En el siguiente ejemplo se muestra la forma correcta de construir una red.

Ejemplo:

En la figura 8 se muestra las actividades predecesoras para un proceso de impresión de un documento.</p>

```
<p align="justify"><br /> </p></div>
```

```
<div id="Layer6">
```

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase=
```

```
"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0"
```

```
width="100%" height="340">
```

```
<param name="movie" value="animaciones/U1L1.swf" />
```

```
<param name="quality" value="high" />
```

```
<embed src="animaciones/U1L1.swf" quality="high"
```

```
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash"
```

```
width="100%" height="340"></embed></object></div>
```

```
<div id="Layer7"><strong>FIGURAS</strong></div>
```

```
<script language="JavaScript1.2">mmLoadMenus();</script>
```

```
<table width="97%" height="130%" align="center"> <tr>
```

```
<td height="78"><div align="center">
```

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase=
```

```
"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0"
```

```
width="98%" height="73">
```

```
<param name="movie" value="animaciones/encabezado.swf" />
```

```
<param name="quality" value="high" />
```

```
<embed src="animaciones/encabezado.swf" quality="high" pluginspage=
```

```
"http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash" width="98%"
```

```
height="73"></embed> </object></div></td></tr>
```

```
<tr> <td width="95%" height="27"><div align="center"><img src=
```

```
"imagenes/modeloboton.png" name="image1" width="17%" height="30" id="image1"
```

```
onmouseover="MM_showMenu(window.mm_menu_0318134401_0,0,29,null,'image1')"
```

```
onmouseout="MM_startTimeout();" /> <a href="glosario.php" target="_blank"></a><a
href="ayuda.htm" target="_blank"></a><a href="pagAcercaDe.php" target="_blank"
onclick="window.open(this.href, this.target, 'width=520,height=420'); return false;"></a></div></td>
</tr>
<tr><td width="95%" height="32" valign="bottom"><div align="center"></div></td></tr>
<tr><td width="95%" height="383" valign="bottom"><div align="center"></div></td></tr>
<tr><td height="19"><div align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase=
"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0"
width="98%" height="31">
<param name="movie" value="animaciones/BotInferiores11.swf" />
<param name="quality" value="high" />
<embed src="animaciones/BotInferiores11.swf" quality="high" pluginspage=
"http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash" width="98%"
height="31"></embed></object></div></td></tr></table>
</body>
</html>

```

Los archivos (o paginad web), que corresponden a las demás lecciones, poseerán una estructura similar, cambiando el contenido, las imágenes y animaciones, según sea el caso.

4.6 EVALUACIÓN DE LA FASE DE ELABORACIÓN

Esta fase nos permitió construir la línea base de la arquitectura del sistema. Para ello, se capturaron los requisitos que habían quedado pendientes de la fase anterior, permitiendo construir el diagrama de casos de uso, los diagramas de clases de análisis y clases de diseño. Al establecer una base de arquitectura sólida, se obtiene una guía para las fases de construcción y transición.

Específicamente, en el flujo de trabajo requisitos, se refina el modelo de casos de uso obtenido durante la fase de inicio, tomando como base los nuevos requisitos funcionales incluidos. En el nuevo modelo se incluye el actor Asistente de Usuario, y se mantienen los actores establecidos en la fase previa, aunque estos, adquieren mayor funcionalidad con la inserción de nuevos casos de uso, que proporcionan mayor consistencia a la arquitectura.

En el flujo de trabajo Análisis, se ajusta el modelo de análisis y de paquetes, de acuerdo al diagrama de casos de usos obtenido una vez se incluyeron los nuevos casos de usos. Esto permitió la construcción de diagramas de clases de análisis y de colaboración que reflejan las relaciones e interacciones entre gestores, interfaces y entidades de los casos de uso “Navegar Contenido de Unidades” y “Configurar Sistema”.

Esta fase se ha concentrado principalmente en el flujo de trabajo diseño, ya que, permitió diseño de la estructura del software para lo cual, se han especificado clases, capas, paquetes e interacción entre los objetos del sistema. Específicamente, se desarrollan los diagramas de clases de diseño para los casos de uso “Navegar Contenido de Unidades” y “Configurar Sistema” para ello se aplicó la tecnología WAE. A partir de esto, se desarrollan los respectivos diagramas de secuencia para los casos de uso, mostrando las interacciones y mensajes entre cada una de las clases de

diseño especificadas para el sistema. Posteriormente, se representan los subsistemas de diseño mediante el diagrama de capas del sistema, estableciendo una organización de los artefactos del modelo de diseño en piezas más manejables. Se desarrollan los diagramas de navegación correspondientes a las actividades del administrador y a las del usuario, estas se representan a través del diagrama de navegación (Modalidad 1: Privado-Administrador) y Diagrama de navegación (Modalidad 2: Privado-Usuario), respectivamente. En cuanto a la naturaleza educativa del software, correspondió a esta fase, el diseño de la interfaz, el cual esta compuesto por el Diseño Educativo, el Diseño Comunicacional y el Diseño Computacional de la aplicación, en este se detallan los objetivos de cada una de las unidades, una selección del contenido que se incluye en el software y las especificaciones de la interfaz. Finalmente, se realiza el diseño de la base de datos, atendiendo a los requisitos del software.

En el flujo de trabajo de implementación se desarrolla el modelo de implementación representado por el diagrama de componentes (parcial) para la navegación del contenido de las unidades. Posteriormente se muestra la implementación de este caso de uso.

4.7 PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN

En esta fase se realizara la codificación del SEMOI enfocándose principalmente en los flujos de trabajo de implementación y pruebas. Su propósito principal es dejar listo la versión operativa inicial del proyecto (versión Beta). Se desarrollarán las interfaces restantes realizando pruebas y ajustes necesarios hasta obtener la versión final. Se integrará y probará la totalidad del sistema.

CAPÍTULO V: FASE DE CONSTRUCCIÓN

5.1 INTRODUCCIÓN

Durante la fase de Elaboración se obtuvo la totalidad de los requisitos funcionales y no funcionales del software, a su vez, se realizó el análisis y el diseño de los mismos.

En la fase de construcción se lleva a cabo la codificación e integración del sistema, para ello se deben haber escogido las herramientas apropiadas para tal fin. Esta fase abarca también las pruebas de todos los componentes del sistema, trabajando principalmente a partir del modelo de diseño obtenido en la fase anterior. Al final de esta fase se tendrá el producto listo en una versión inicial (beta), totalmente operativa.

La fase de Construcción, se desarrolla mediante los flujos de trabajo implementación y pruebas, debido a que la totalidad de los requisitos han sido capturados en fases anteriores, y no son necesarias nuevas iteraciones, en los flujos de trabajo requisitos, análisis o diseño.

Para el flujo de trabajo de implementación, se completará el desarrollo del diagrama de componentes realizado parcialmente en la fase de elaboración.

En el flujo de trabajo Pruebas se planifican, diseñan y ejecutan las pruebas de unidad y de integración del sistema, esto permitirá comprobar de forma efectiva el correcto funcionamiento de la aplicación. Posteriormente, se implementarán las clases que corresponden a los nuevos componentes incluidos en el diagrama y a los cuales se le realizaron pruebas de unidad e integración. Se implementan los objetos definidos en el diseño educativo, comunicacional y computacional, adecuándose a las

posibilidades de la herramienta de desarrollo que se va a utilizar y se implementan las interfaces restantes.

5.2 FLUJO DE TRABAJO: IMPLEMENTACIÓN

Este flujo de trabajo engloba la mayor parte del trabajo de la fase de construcción, y permite la implementación de los resultados obtenidos en el diseño, en términos de componentes. El resultado de este flujo proporcionara un software completo listo para ser sometido a pruebas, obteniéndose allí la aplicación final.

En este flujo se amplía el modelo de implementación desarrollado en la fase de elaboración, para lo cual se incluye las funcionalidades correspondientes al caso de uso Configurar Sistema.

5.2.1 Escogencia de las Herramientas de Desarrollo

Para la implementación del SEMOI, se utilizan las siguientes herramientas: El lenguaje de programación PHP, que es el lenguaje de servidor que permitirá la codificación de las páginas del servidor, el establecimiento de sesiones de usuario, consultas a la base de datos, entre otras funcionalidades. Php es ideal para la creación de paginas Web, ya que, es un potente lenguaje que soporta la programación orientada a objetos, puede estar incrustado en código HTML, dispone de librerías de conexión con la mayoría de sistemas de gestión de base de datos para almacenamiento de información permanente en el servidor, además, proporciona soporte a múltiples protocolos de comunicaciones en Internet, en caso de ser necesarios. En la Fig. 5.1 se observa un ejemplo de implementación de código php, a través de la interfaz del software Macromedia Dreamweaver 8. Adicionalmente, se hace necesaria la inclusión de código HTML para la implementación de algunas interfaces del software. Para ello se utiliza, igualmente Macromedia Dreamweaver 8.

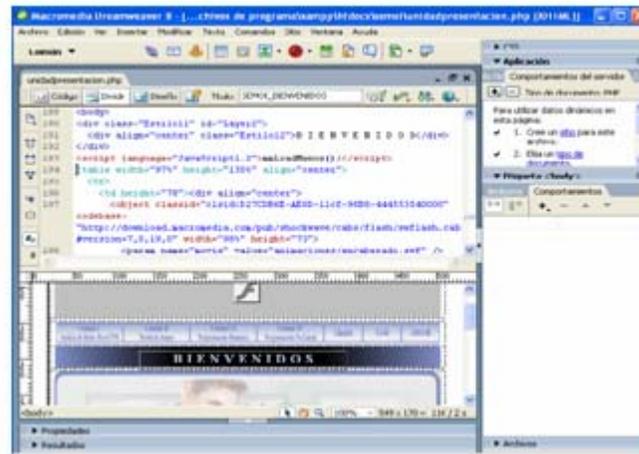


Fig. 5.1. Interfaz del Entorno de Programación PHP en Dreamweaver.

Para el diseño de las interfaces de usuarios se utiliza Macromedia Flash 8, que proporciona asistencia para la creación de contenidos gráficos, animaciones y aplicaciones gestionadas por datos. Este software resulta esencial para el desarrollo de los micromundos interactivos, juegos y/o actividades didácticas incluidas en el SEMOI. Esta herramienta incluye Macromedia Flash Player, el cual es un reproductor que garantiza que todo el contenido flash (extensión .swf) pueda visualizarse y esté disponible en todos los navegadores, plataformas y escritorios compatibles con Internet. La interfaz de flash se observa en la Fig 5.2.

Otra herramienta considerada esencial para el desarrollo de gráficos e imágenes es Macromedia Fireworks 8, que permite la creación de interfaces de usuario atractivas basadas en las facilidades y ventajas que ofrece. En la Fig. 5.3 se muestra la interfaz de Fireworks.



Fig. 5.2 Interfaz de Macromedia Flash 8.

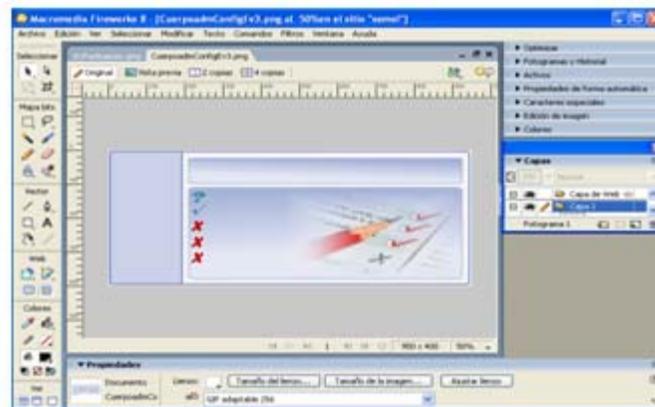


Fig. 5.3 Interfaz de Macromedia Fireworks 8.

Todas estas herramientas permiten a los desarrolladores la creación de sitios Web interesantes, completos y robustos. Por ello, son escogidos para la realización del SEMOI.

5.2.2 Escogencia del Gestor de Base de Datos

Para la construcción de la base de datos del SEMOI se escoge el manejador MySQL, debido a que proporciona facilidades de conexión con el lenguaje PHP, garantiza la integridad y seguridad de los datos, aprovecha la potencia de sistemas

multiprocesador, ya que, ofrece características multihilo, soporta gran cantidad de datos para cada columna, ofrece gran portabilidad entre sistemas, es multiusuario, entre otras. La interfaz del XAMP que permite la creación de la base de datos MySQL se muestra en la Fig 5.4.

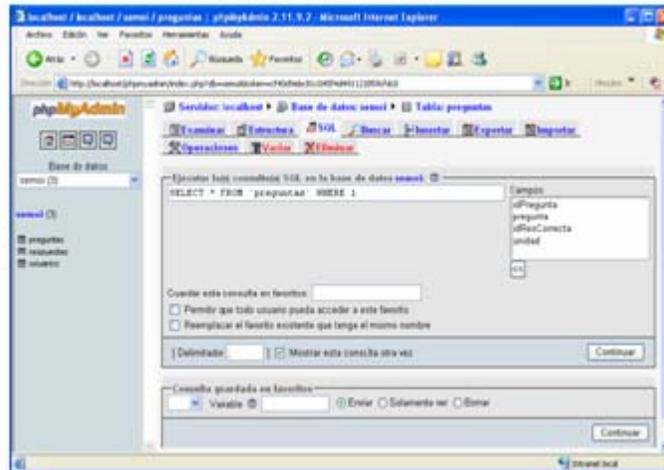


Fig. 5.4 Interfaz del entorno MySql.

5.2.3 Construcción de la Base de Datos

La estructura básica de la base de datos del SEMOI, fue realizada en la sección 4.4.2 Diseño de la Base de Datos del capítulo anterior, en esta se mostró el diagrama entidad – relación, que permitió representar relaciones entre las tablas y, posteriormente, se realizó una breve descripción de los atributos de las tablas.

Debido a que no se registran cambios en la estructura de la base de datos, en esta sección se expone la implementación de la base de datos creada, a través del XAMP.

A continuación se muestran las consultas necesarias para la creación de la base de datos.

Creación de la base de datos:

```
CREATE DATABASE 'semoi';
```

Creación de la tabla pregunta:

```
CREATE TABLE 'semoi'.preguntas' (
'idPregunta' INT( 3 ) NOT NULL ,
'pregunta' VARCHAR( 500 ) NOT NULL ,
'idResCorrecta' INT( 4 ) NOT NULL ,
'unidad' INT( 1 ) NOT NULL ,
PRIMARY KEY ( 'idPregunta' )
CONSTRAINT 'Fk_preguntas_1' FOREIGN KEY 'FK_ preguntas_1' ('idResCorrecta')
REFERENCES respuesta ('idRespuesta') ON DELETE RESTRICT ON UPDATE
RESTRICT) ENGINE = MYISAM
```

Creación de la tabla respuestas:

```
CREATE TABLE 'semoi'.respuestas' (
'idRespuesta' INT( 4 ) NOT NULL ,
'Respuesta' VARCHAR( 500 ) NOT NULL ,
'idPregunta' VARCHAR( 3 ) NOT NULL ,
PRIMARY KEY ( 'idRespuesta' )
) ENGINE = MYISAM
```

Creación de la tabla usuarios:

```
CREATE TABLE 'semoi'.usuarios' (
'id' INT( 4 ) NOT NULL ,
'usuarios' VARCHAR( 20 ) NOT NULL ,
```

```

'contraseña' VARCHAR( 8 ) NOT NULL ,
'nivel' INT( 1 ) NOT NULL ,
'nombre' VARCHAR( 20 ) NOT NULL ,
'apellido' VARCHAR( 20 ) NOT NULL ,
'correo' VARCHAR( 20 ) NOT NULL ,
PRIMARY KEY ( 'id' )
) ENGINE = MYISAM

```

En la figura 5.5, se muestra el diagrama de vistas del sistema proporcionado por el manejador de la base de datos, donde se especifican las tablas, sus atributos, tipos y relaciones.

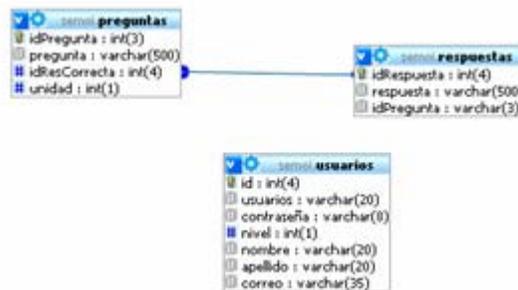


Fig. 5.5. Diagrama de Vistas de SEMOI.

5.2.4 Modelo de Implementación

Para esta fase se agrega funcionalidad al software, por lo cual se hace necesario rediseñar el modelo de implementación mostrado en el capítulo anterior. De acuerdo a los cambios, e interfaces incluidas al software se construye un modelo de implementación que engloba todas las funciones del software. Para su representación se utiliza el diagrama de componentes de la Figura 5.6. En este diagrama se muestran todos los componentes requeridos para la implementación del sistema.

Se puede observar que el diagrama muestra todos los componentes del sistema. Estos se encuentran encapsulados en paquetes lo que agiliza la navegación a través del software.

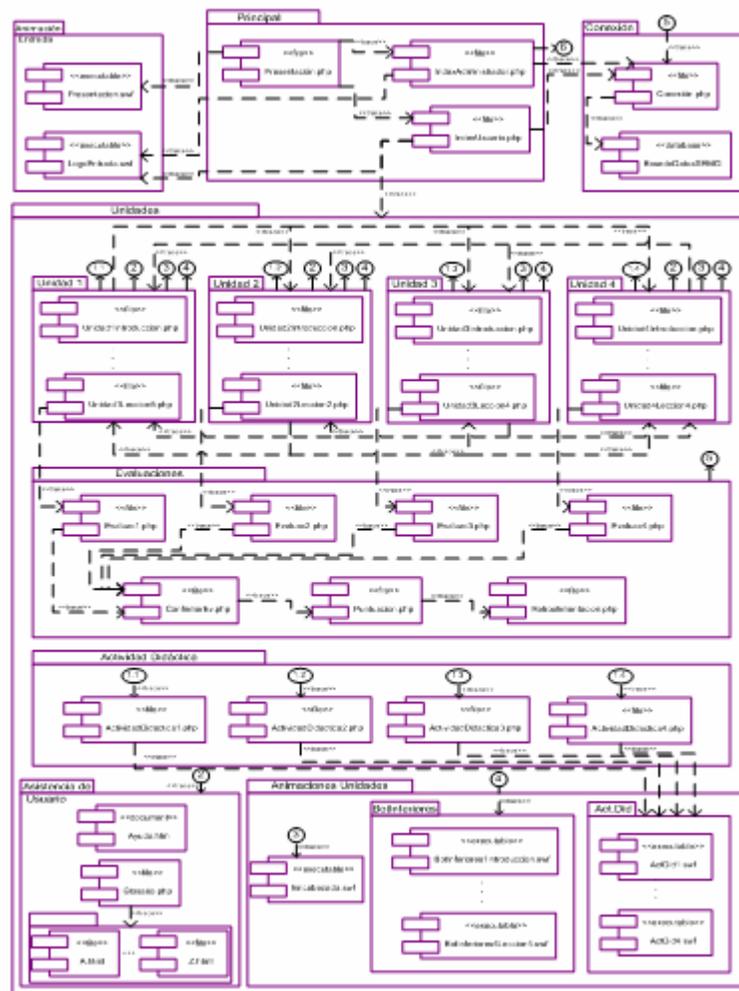


Fig. 5.6. Diagrama de Componentes del Sistema (1/2)

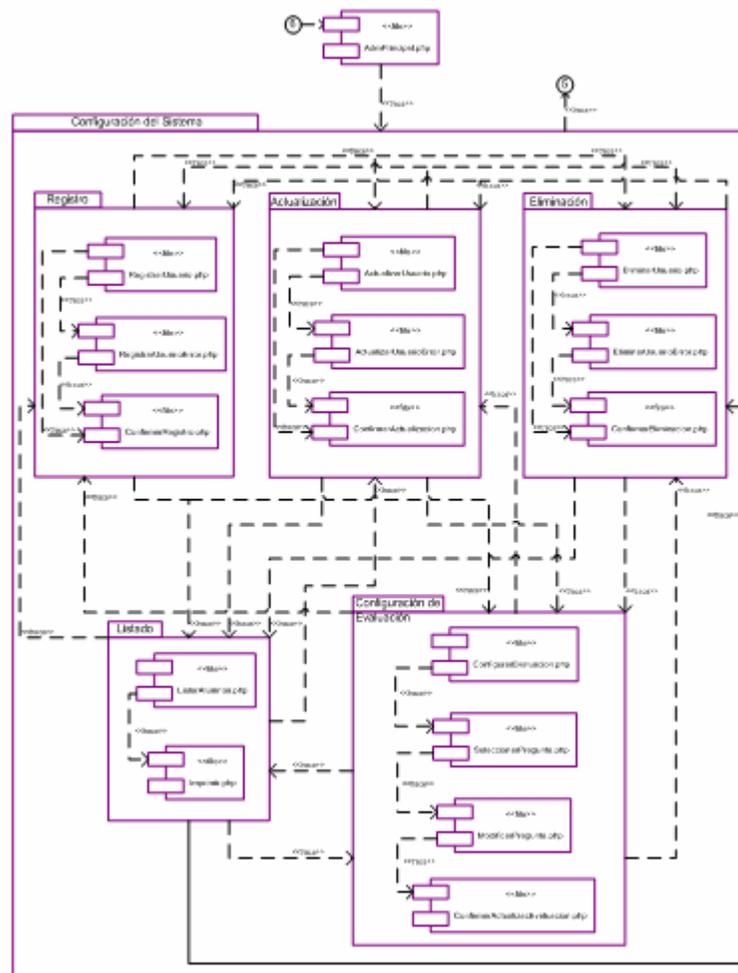


Fig. 5.6. Diagrama de Componentes del Sistema (2/2)

5.3 FLUJO DE TRABAJO: PRUEBAS

En este flujo de trabajo se comprueba el resultado de la implementación de los componentes. Su objetivo fundamental es validar el software, con la intención de encontrar errores, para luego depurarlo y obtener como resultado un software de calidad. De esta manera, se verificará el funcionamiento de cada componente, la comunicación entre ellos y se validan las entradas proporcionadas por diferentes usuarios.

Se realizarán pruebas de unidad (caja negra) y de integridad del sistema, tomando como base el diagrama de componentes mostrado previamente, de esta manera se determinará si el software cumple con los requisitos establecidos.

5.3.1 Pruebas de Unidad

Las pruebas de unidad del SEMOI se realizan mediante el modelo de caja Negra. Para ello, se debe identificar el conjunto de valores que pueden ser ingresados por el usuario. Estos valores se presentan como clases de equivalencia, para asegurar que la totalidad de los casos probables se incluyen en las pruebas. Para cada valor se evalúa si ofrece una respuesta acorde a lo requerido.

Para la implementación de las pruebas, se han escogido los casos de uso Registrar Usuario y Configurar Evaluación. A través de estos, se mostrarán las respuestas del sistema a las entradas del usuario.

En la tabla 5.1 se muestran las clases de equivalencias para el caso de uso Registrar Usuario. Para este caso de uso el administrador del sistema debe incluir un nuevo usuario en el sistema (base de datos).

Tabla 5.1. Clases de Equivalencia para el caso de uso Registrar Usuario (1/3)

N°	Dato	Clase de Equivalencia	Válido	No válido
1	Usuario	Caracteres Numéricos	x	
2	Usuario	Caracteres Alfabéticos	x	
3	Usuario	Caracteres Alfanuméricos	x	
4	Usuario	Longitud de caracteres < 3		x
5	Usuario	Longitud de caracteres > 20		x

6	Usuario	3<=Long. de caracteres<=20	x	
7	Usuario	Dato nulo		x
8	Usuario	Dato repetido		x
9	Contraseña	Caracteres Numéricos	x	
10	Contraseña	Caracteres Alfabéticos	x	
11	Contraseña	Caracteres Alfanuméricos	x	
12	Contraseña	Longitud de caracteres < 4		x
13	Contraseña	Longitud de caracteres > 8		x
14	Contraseña	4<=Long. de caracteres<= 8	x	
15	Contraseña	Dato nulo		x
16	Nombre estudiante	Caracteres Numéricos		x
17	Nombre estudiante	Caracteres Alfabéticos	x	
18	Nombre estudiante	Caracteres Alfanuméricos		x
19	Nombre estudiante	Longitud de caracteres < 3		x
20	Nombre estudiante	Longitud de caracteres > 20		x
21	Nombre estudiante	3<=Long. de caracteres<= 20	x	
22	Nombre estudiante	Dato nulo		x
23	Apellido estudiante	Caracteres Numéricos		x

Tabla 5.1. Clases de Equivalencia para el caso de uso Registrar Usuario (2/3)

N°	Dato	Clase de Equivalencia	Válido	No válido
24	Apellido estudiante	Caracteres Alfabéticos	x	
25	Apellido estudiante	Caracteres Alfanuméricos		x
26	Apellido estudiante	Longitud de caracteres < 3		x
27	Apellido estudiante	Longitud de caracteres > 20		x
28	Apellido estudiante	3<=Long. de caracteres<= 20	x	

29	Apellido estudiante	Dato nulo		x
30	Correo electrónico	Caracteres Numéricos	x	
31	Correo electrónico	Caracteres Alfabéticos	x	
32	Correo electrónico	Caracteres Alfanuméricos	x	
33	Correo electrónico	Longitud de caracteres < 5		x
34	Correo electrónico	Longitud de caracteres >35		x
35	Correo electrónico	5<=Long. de caracteres<= 35	x	
36	Correo electrónico	Dato nulo		x
37	Correo electrónico	Carencia del símbolo '@'		x
38	Correo Electrónico	Carencia del símbolo '.' antes del dominio del correo		x
39	Correo electrónico (nombre de Correo)	Inclusión obligatoria de caracteres antes del '@'	x	
40	Correo electrónico (extensión)	Inclusión obligatoria de caracteres entre el '@' y el '.'	x	
41	Correo electrónico (Dominio)	Inclusión obligatoria de caracteres después del '.'	x	
42	Correo electrónico	Carácter "espacio en blanco" (' ')		x

Tabla 5.1. Clases de Equivalencia para el caso de uso Registrar Usuario (3/3)

N°	Dato	Clase de Equivalencia	Válido	No válido
43	Nivel	Caracteres Numéricos	x	
44	Nivel	Caracteres Alfabéticos		x
45	Nivel	Caracteres Alfanuméricos		x
46	Nivel	Longitud de caracteres >1		x
47	Nivel	Longitud de caracteres = 1	x	

48	Nivel	Dato nulo		x
49	Nivel	Valor = 1	x	
50	Nivel	Valor = 2	x	
51	Nivel	Valor > 2 valor < 0		x

Para verificar que el sistema funciona correctamente se plantean casos de pruebas, con la información que podría ser ingresada por los usuarios. De esta manera, se podrá observar las respuestas que genera la aplicación ante la entrada proporcionada por el administrador del sistema. En la tabla 5.2 se presentan algunos casos de prueba de caja negra, correspondientes al caso de uso Registrar Usuario.

Tabla 5.2. Casos de prueba de caja negra para Registrar Usuario (1/3)

N°	Dato	Caso de Prueba	Salida	Clases cubiertas
1	Usuario	1112	Válido	1,6
2	Usuario	María	Válido	2,6
3	Usuario	juan40	Válido	3,6
4	Usuario	D3	Inválido	4
5	Usuario	Maria Alejandra del Sol	Inválido	5
6	Usuario	(Null)	Inválido	7

Tabla 5.2. Casos de prueba de caja negra para Registrar Usuario (2/3)

N°	Dato	Caso de Prueba	Salida	Clases cubiertas
7	Usuario	María	Inválido	8
8	Contraseña	17235467	Válido	9,14
9	Contraseña	Mariíta	Válido	10,14
10	Contraseña	Martin2@	Válido	11,14

11	Contraseña	lis	Inválido	12
12	Contraseña	marianita	Inválido	13
13	Contraseña	(Null)	Inválido	15
14	Nombre estudiante	6465675	Inválido	16
15	Nombre estudiante	Maria Alejandra	Válido	17,21
16	Nombre estudiante	María Alejandra23	Inválido	18
17	Nombre estudiante	li	Inválido	19
18	Nombre estudiante	Lisandro Martinez Perez	Inválido	20
19	Nombre estudiante	(Null)	Inválido	22
20	Apellido estudiante	4355 64366 35567 3455	Inválido	23,27
21	Apellido estudiante	Martínez Parra	Válido	24,28
22	Apellido estudiante	Mata21	Inválido	25
23	Apellido estudiante	ze	Inválido	26
24	Apellido estudiante	(Null)	Inválido	29
25	Correo electrónico	Reyna01@gmail.com	Válido	32,35,39,40,41
26	Correo electrónico	franciscoGonzalopez MataHotmail.com.ve	Inválido	34,37

Tabla 5.2. Casos de prueba de caja negra para Registrar Usuario (3/3)

N°	Dato	Caso de Prueba	Salida	Clases cubiertas
27	Correo electrónico	n@	Inválido	33
28	Correo electrónico	(Null)	Inválido	36
29	Correo electrónico	Salazar@hotmail	Invalido	38
30	Correo electrónico	aiaemrl@hotmail.com	Válido	31,32,35,39,40,41
31	Correo electrónico	Maria @hotmail.com	Invalido	42
32	Nivel	1	Válido	43,47,49

33	Nivel	uno	Inválido	44,46
34	Nivel	@	Inválido	45
35	Nivel	(Null)	Inválido	48
36	Nivel	2	Válido	50
37	Nivel	5	Inválido	51

En la tabla 5.3 se observan las clases de equivalencia para el caso de uso Configurar Evaluación. En este, el administrador debe insertar la unidad que corresponde a la evaluación que se quiere modificar, posteriormente, se elige la pregunta a modificar, y luego, se realizará la inserción de la pregunta y de las respuestas correspondientes.

Tabla 5.3. Clases de Equivalencia para el caso de uso Configurar Evaluación(1/3)

Nº	Dato	Clase de Equivalencia	Valido	No valido
1	Unidad	Caracteres Numéricos	x	
2	Unidad	Caracteres Alfabéticos		x
3	Unidad	Caracteres Alfanuméricos		x
4	Unidad	Longitud de caracteres = 1	x	

Tabla 5.3. Clases de Equivalencia para el caso de uso Configurar Evaluación(2/3)

Nº	Dato	Clase de Equivalencia	Valido	No valido
5	Unidad	1 > Long. de caracteres > 1		x
6	Unidad	1 > Valor > 4		x
7	Unidad	Dato Nulo		x
8	Nueva Pregunta	Caracteres Numéricos	x	
9	Nueva Pregunta	Caracteres Alfabéticos	x	

10	Nueva Pregunta	Caracteres Alfanuméricos	x	
11	Nueva Pregunta	Longitud de caracteres < 5		x
12	Nueva Pregunta	Long. de caracteres > 500		x
13	Nueva Pregunta	5<=Long de caract.<=500	x	
14	Nueva Pregunta	Dato Nulo		x
15	Respuesta Correcta	Caracteres Numéricos	x	
16	Respuesta Correcta	Caracteres Alfabéticos	x	
17	Respuesta Correcta	Caracteres Alfanuméricos	x	
18	Respuesta Correcta	Longitud de caracteres < 2		x
19	Respuesta Correcta	Long. de caracteres > 500		x
20	Respuesta Correcta	2<= Longitud de caracteres <= 500	x	
21	Respuesta Correcta	Dato Nulo		x
22	Respuesta Incorrecta1	Caracteres Numéricos	x	
23	Respuesta Incorrecta1	Caracteres Alfabéticos	x	
24	Respuesta Incorrecta1	Caracteres Alfanuméricos	x	
25	Respuesta Incorrecta1	Longitud de caracteres < 2		x
26	Respuesta Incorrecta1	Long. de caracteres > 500		x

Tabla 5.3. Clases de Equivalencia para el caso de uso Configurar Evaluación(3/3)

N°	Dato	Clase de Equivalencia	Valido	No valido
27	Respuesta Incorrecta1	2<= Longitud de caracteres <= 500	x	
28	Respuesta Incorrecta1	Dato Nulo		x
29	Respuesta Incorrecta2	Caracteres Numéricos	x	

30	Respuesta Incorrecta2	Caracteres Alfabéticos	x	
31	Respuesta Incorrecta2	Caracteres Alfanuméricos	x	
32	Respuesta Incorrecta2	Longitud de caracteres < 2		x
33	Respuesta Incorrecta2	Long. de caracteres > 500		x
34	Respuesta Incorrecta2	2<= Longitud de caracteres <= 500	x	
35	Respuesta Incorrecta2	Dato Nulo		x
36	Respuesta Incorrecta3	Caracteres Numéricos	x	
37	Respuesta Incorrecta3	Caracteres Alfabéticos	x	
38	Respuesta Incorrecta3	Caracteres Alfanuméricos	x	
39	Respuesta Incorrecta3	Longitud de caracteres < 2		x
40	Respuesta Incorrecta3	Long. de caracteres > 500		x
41	Respuesta Incorrecta3	2<= Longitud de caracteres <= 500	x	
42	Respuesta Incorrecta3	Dato Nulo		x

En la tabla 5.4, se presentan las pruebas de caja negra para el caso de uso Configurar Evaluación. A través, de estas pruebas se comprobarán las clases de equivalencia mostradas anteriormente.

Tabla 5.4. Casos de prueba de caja negra para Configurar Evaluación

Nº	Dato	Caso de Prueba	Salida	Clases Cubiertas
1	Unidad	3	Válido	1,4
2	Unidad	uno	Inválido	2,5
3	Unidad	34e	Inválido	3,5
4	Unidad	5	Inválido	6
5	Unidad	(Null)	Invalido	7

6	Nueva Pregunta	¿Una red debe poseer 2 nodos?	Válido	8,9,10,13
7	Nueva Pregunta	Que?	Invalido	11
8	Nueva Pregunta	(Null)	Inválido	14
9	Respuesta Correcta	Puede tener 2 o más nodos!	Válido	15,16,17,20
10	Respuesta Correcta	N	Inválido	18
11	Respuesta Correcta	(Null)	Inválido	21
12	Respuesta Incorrecta1	No debe tener 2 nodos!	Válido	22,23,24,27
13	Respuesta Incorrecta1	s	Inválido	25
14	Respuesta Incorrecta1	(Null)	Inválido	28
15	Respuesta Incorrecta2	Solo debe tener 5 nodos!	Válido	29,30,31,34
16	Respuesta Incorrecta2	2	Inválido	32
17	Respuesta Incorrecta2	(Null)	Inválido	35
18	Respuesta Incorrecta3	Si debe tener 2 nodos!	Válido	36,37,38,41
19	Respuesta Incorrecta3	x	Inválido	39
20	Respuesta Incorrecta3	(Null)	Inválido	42

5.3.2 Pruebas de Integridad

Las pruebas de integración se utilizan para verificar que los componentes interactúen entre si de la forma correcta, una vez hayan sido integrados. Estas se basan principalmente en los diagramas de secuencia, en los cuales, se muestran sucesivamente, las acciones que puede realizar el sistema. [Ver capítulo IV. Apartado 4.4.1.3.1.2. Diagramas de Secuencia]

Básicamente, se probará la comunicación entre cada uno de los componentes desarrollados y probados individualmente. Se verificará si el sistema es capaz de mostrar las interfaces correspondientes a cada caso y si siguen la secuencia apropiada.

La integración de los componentes ha sido desarrollada por fases. Estas fases se muestran en la Fig 5.7, a través del diagrama de componentes del sistema.

Evidentemente, se realizarán pruebas de integridad correspondientes a todas las fases (mostradas en el diagrama de componentes de la Fig. 5.7), hasta llegar a la integración y depuración total del sistema. Sin embargo, se mostrará, únicamente, la integración de los componentes correspondientes a la fase 4, identificados en el diagrama de la Figura 5.7. Posteriormente, se mostrará parcialmente la integración de los componentes de la fase 13.

5.3.2.1 Integración del caso de uso Registrar Usuario

Este caso de prueba verifica la funcionalidad de la implementación del registro del usuario. La secuencia se inicia desde cualquier interfaz del administrador que posea el link a Registrar Usuario, se toma como ejemplo la página principal del administrador (Ver Fig. 5.8). Posteriormente, se accederá a la interfaz del Registro.

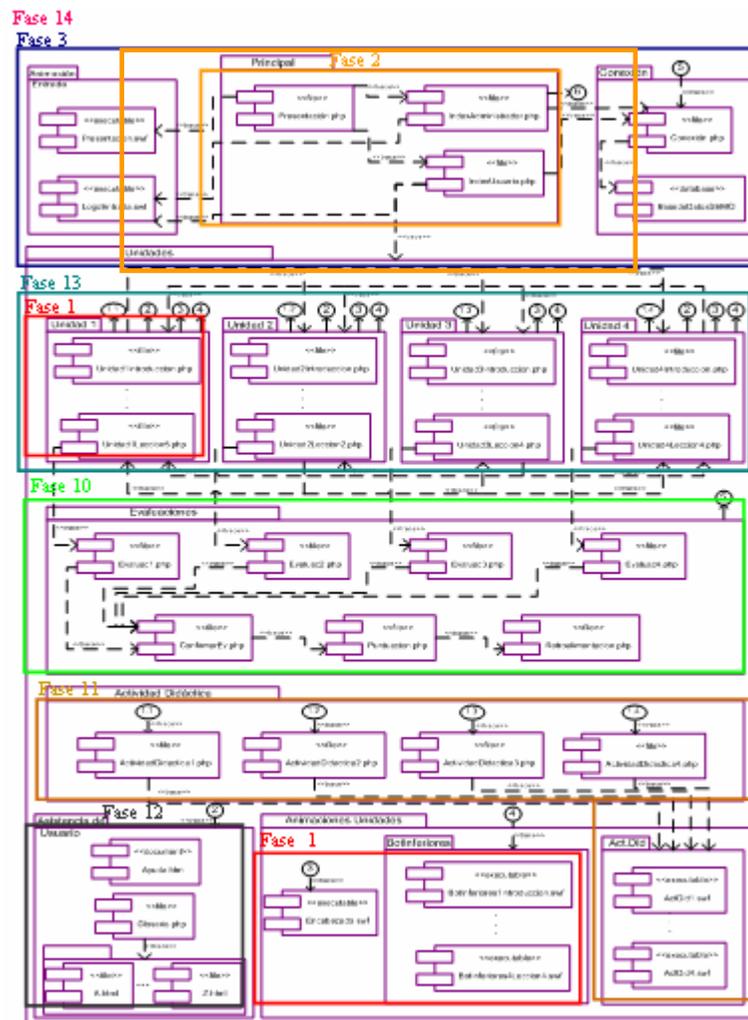


Fig. 5.7. Identificación de fases de Integración a través del Diagrama de Componentes del Sistema (1/3)

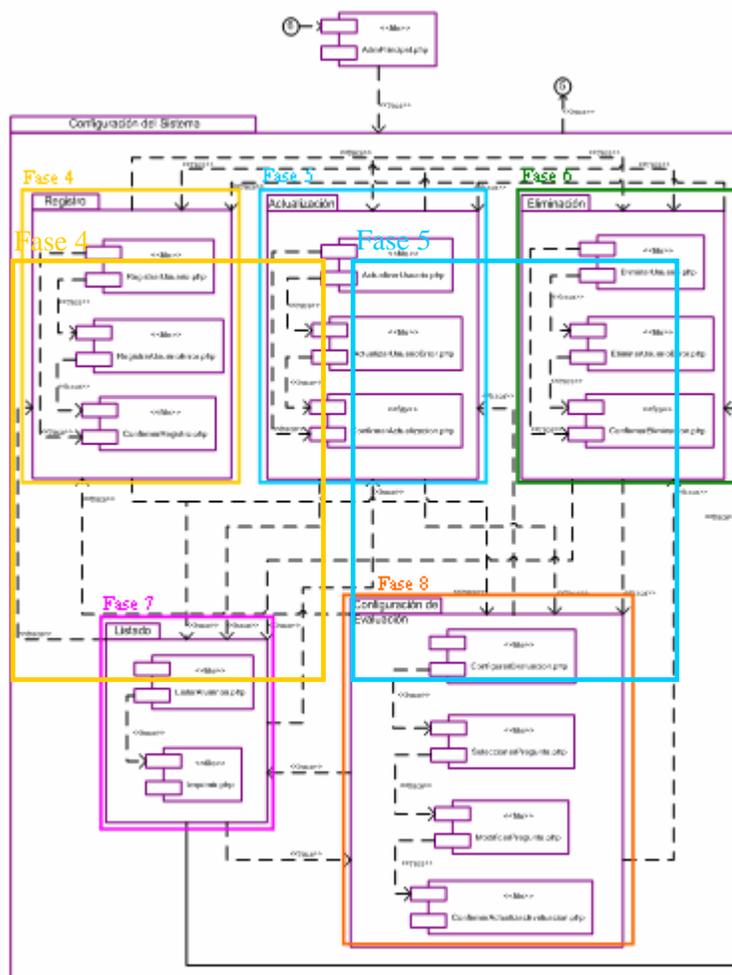


Fig. 5.7. Identificación de fases de Integración a través del Diagrama de Componentes del Sistema (2/3)

Leyenda:

<input type="checkbox"/>	Fase 1	<input type="checkbox"/>	Fase 2
<input type="checkbox"/>	Fase 3	<input type="checkbox"/>	Fase 4
<input type="checkbox"/>	Fase 5	<input type="checkbox"/>	Fase 6
<input type="checkbox"/>	Fase 7	<input type="checkbox"/>	Fase 8
<input type="checkbox"/>	Fase 9	<input type="checkbox"/>	Fase 10
<input type="checkbox"/>	Fase 11	<input type="checkbox"/>	Fase 12
<input type="checkbox"/>	Fase 13	<input type="checkbox"/>	Fase 14

Fig. 5.7. Identificación de fases de Integración a través del Diagrama de componentes del sistema (3/3)



Fig. 5.8. Pagina principal administrador (Registrar Usuario).

Datos de Entrada: Usuario, Contraseña, Nombre de Estudiante, Apellido de Estudiante, Correo Electrónico, Nivel de acceso. (Ver tabla 5.5)

Tabla 5.5. Datos de entrada para verificar la integración de Registrar Usuario

N°	Dato	Caso de Prueba
1	Usuario	marta20
2	Contraseña	123456
3	Nombre de Estudiante	Marta
4	Apellido de Estudiante	Perez
5	Correo electrónico	perez@hotmail.com
6	Nivel de Acceso	2

Resultado: Obtención de la interfaz de usuario correspondiente.

Condiciones para el registro: El usuario, el nombre y el apellido del estudiante deben tener entre 3 y 20 caracteres. El nombre y el apellido, solo aceptan caracteres alfabéticos (se permiten caracteres en blanco “ ”). La contraseña, a su vez, debe estar compuesta por entre 4 y 8 caracteres. Para el campo de correo electrónico, se debe establecer el signo ‘@’, el cual debe tener caracteres antes y después del mismo, siendo el número de caracteres permitidos entre 5 y 35. El nivel de acceso debe establecerse, únicamente, en ‘1’, para privilegios de administrador, o ‘2’, para privilegios de usuario. Adicionalmente, no se podrá insertar un usuario que ya se haya registrado en el sistema con anterioridad.

Procedimiento de prueba:

- Activar la interfaz de login del administrador.
- Activar la interfaz de administración (mostrada en la Fig.5.8) o cualquier otra interfaz de administrador que proporcione acceso al Registro del Usuario.
- Acceder a la interfaz de registro de usuario e introducir los datos de entrada. Se toma como ejemplo los de la tabla 5.5.(Ver Fig.5.9)

- Una vez se presiona el botón Enviar el sistema valida los campos de entrada. Durante este proceso se pueden dar 3 casos:
 1. Si los campos no cumplen con las condiciones establecidas (no nulidad, cantidad de caracteres, empleo de @ para el correo, entre otras), se podrá apreciar un mensaje como el de la Fig. 5.10. Posteriormente, se redireccionará a la pagina de Registro de Usuario nuevamente.
 2. Si todos los campos cumplen con las condiciones establecidas se realiza una búsqueda en la base de datos con el fin de comprobar que no existe algún usuario igual al que está siendo ingresado (en este caso, 'marta20'). Si no se encuentra este usuario registrado, se activa el gestor que ejecuta la consulta para insertar al usuario en la base de datos (Ver Fig. 5.11). Luego, se envía a la interfaz de confirmación de Registro de usuario (Ver Fig. 5.12). Para comprobar la inserción de del nuevo usuario ('marta20') se realiza la consulta a la base de datos, mostrada en la Figura 5.13.
 3. En caso de que ya exista ese usuario registrado ('marta20'), se dirige a una interfaz que muestra un mensaje de error solicitando un nuevo nombre de usuario (Ver Fig. 5.14). Igualmente, esta permitirá el registro del usuario.

Fig. 5.9. Introducción de datos en la interfaz de Registro de Usuario



Fig. 5.10. Mensaje de Error

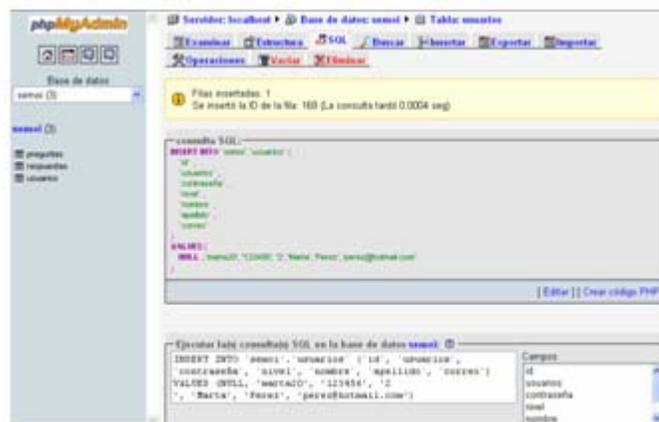


Fig. 5.11. Inserción de usuario en la base de datos



Fig. 5.12. Confirmación de Registro de Usuario.

phpMyAdmin

Servidor: localhost • Base de datos: sesesi • Tabla: usuarios

Operaciones: Ejecutar SQL, Buscar, Mostrar, Exportar, Importar, Operaciones, Vaciar, Eliminar

Mostrando registros 0 - 6 (7 total, La consulta tardó 0.0004 seg)

Consulta SQL:

```
SELECT *
FROM usuarios
LIMIT 0, 30
```

Mostrar 30 filas empezando de 0 en modo horizontal y repetir los encabezados cada 100 columnas

Organizar según la clave: Ninguna

	ID	usuario	contraseña	nivel	nombre	apellido	correo
<input type="checkbox"/>	55	administrador	1234	1	administrador	cya	admin@hotmail.com
<input type="checkbox"/>	57	ADMIN	1122	1	administrador	Sistema	admin@hotmail.com
<input type="checkbox"/>	149	francisco	felpe	2	francisco	valezar	fr@fr
<input type="checkbox"/>	158	sergio	sergi	2	sergio	axaxax	sd@sd
<input type="checkbox"/>	160	juan40	vzcx	2	scw	wb	oA2d@fr
<input type="checkbox"/>	162	maria	1452783	1	cinche	ren	lind@fr
<input type="checkbox"/>	170	maria20	123456	2	Maria	Perez	perez@hotmail.com

Fig. 5.13. Consulta de confirmación de Registro de Usuario.

SEPRO Software Educativo de Modelos de Operaciones I Sesión: Administrador

Seleccione la Opción de Administración de Su Preferencia

Registrar Usuario, Actualizar Usuario, Eliminar Usuario, Consultar Usuarios, Eliminar Usuarios, Cerrar Sesión

Registrar Nuevo Usuario

Usuario:

Contraseña:

Nombre del Estudiante:

Apellido del Estudiante:

Correo Electrónico:

Nivel de Acceso*:

*Nivel: 1 por Defecto de Administrador 2 por Defecto de Usuario

Enviar

Actualmente existe un usuario registrado con ese nombre. Por Favor, ingrese otro nombre de usuario

Fig. 5.14. Interfaz de Error.

Código Fuente del caso de uso Registrar Usuario

El código presentado a continuación corresponde a la implementación del caso de uso registra usuario, al cual se le han realizado pruebas de unidad e integración en apartados anteriores.

Página Registrar Usuario: (pagAdministradorRegistrarUsuario.php)

```

<?php
if (!isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "1";
$MM_donotCheckaccess = "false";
// *** Restricción de acceso a la pagina
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    $isValid = False;
    if (!empty($UserName)) {
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && false) {
            $isValid = true;
        }
    }
    return $isValid;
}
$MM_restrictGoTo = "indexerroradministrador.php";
if (!(isset($_SESSION['MM_Username']) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_QUERY_STRING) && strlen($_QUERY_STRING) > 0)
        $MM_referrer .= "?" . $_QUERY_STRING;

```



```

        { p=val.indexOf('@');
        if (p<1 || p==(val.length-1)) errors+="- "+nm+": Debe contener una direcci3n de correo
electr3nico.\n\n';
        }
        else if (test!=='R')
        {
        num = parseFloat(val);
        if (isNaN(val)) errors+="- "+nm+": Debe contener un N3mero.\n\n';
        if (test.indexOf('inRange') != -1)
        { p=test.indexOf(':');
        min=test.substring(8,p); max=test.substring(p+1);
        if (num<min || max<num) errors+="- "+nm+": Debe Contener un n3mero entre '+min+' y
'+max+'.\n\n';
        } } }
        else if (test.charAt(0) == 'R') errors += ' - '+nm+": No debe ser NULO.Ingrese el valor
correspondiente. \n\n';
        }
    }
    if (errors) alert('ERROR:\n\n'+errors);
    document.MM_returnValue = (errors == "");
}
//-->
</script>
</head>
<body>
<div id="Layer2">
<table width="100%" height="100%">
<tr><td width="20%" valign="top"><div align="center" class="Estilo2"><br />
<span class="Estilo3">Seleccione la Opci3n de Administraci3n de Su Preferencia
</span></div>
<p align="center"><br />
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">

```

```

    <param name="movie" value="animaciones/button8.swf" />
    <param name="quality" value="high" />
  <param name="bgcolor" value="#CBD1E9" />
  <embed src="animaciones/button8.swf" quality="high"
  pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
  waveFlash" type="application/x-shockwave-flash" width="120" height="28"
  bgcolor="#CBD1E9"></embed>
</object>
</p><p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/ActualizarUsuario.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/ActualizarUsuario.swf" quality="high"
  pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
  waveFlash" type="application/x-shockwave-flash" width="120" height="28"
  bgcolor="#CBD1E9"></embed>
</object>
</p><p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="eliminar.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="eliminar.swf" quality="high"
  pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
  waveFlash" type="application/x-shockwave-flash" width="120" height="28"
  bgcolor="#CBD1E9"></embed>
</object></p>
<p align="center">

```

```

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
  <param name="movie" value="animaciones/confEv.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#CBD1E9" />
  <embed src="animaciones/confEv.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed>
</object> </p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
  <param name="movie" value="animaciones/explorarUnidades.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#CBD1E9" />
  <embed src="animaciones/explorarUnidades.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed>
</object></p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
  <param name="movie" value="animaciones/listarAlumnos.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#CBD1E9" />
  <embed src="animaciones/listarAlumnos.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed></object></p></td></tr></table></div>

```

```

<div id="Layer3">
<form action="pagAdministradoConfirmarRegUsuario.php" method="post" name="form1"
id="form1"
onsubmit="MM_validateForm('Usuario','R','NombreDeEstudiante','R','ApellidoDeEstudiante','R','
CorreoElectronico','RisEmail','Nivel','RinRange1:2','Contrasena','R');return
document.MM_returnValue">
<table width="100%" height="100%"><tr>
<td><strong>Usuario</strong></td>
<td><label>
<input name="Usuario" type="text" id="Usuario" style="background:center"/>
</label></td> </tr>
<tr><td><strong>Contrase&ntilde;a</strong></td><td><label>
<input name="Contrasena" type="password" id="Contrasena" style="background:center" />
</label></td> </tr><tr>
<td><strong>Nombre del Estudiante </strong></td>
<td><label>
<input name="NombreDeEstudiante" type="text" id="NombreDeEstudiante"
style="background:center"/>
</label></td></tr>
<tr><td><strong>Apellido del Estudiante </strong></td>
<td><label>
<input name="ApellidoDeEstudiante" type="text" id="ApellidoDeEstudiante"
style="background:center"/>
</label></td></tr>
<tr><td><strong>Correo Electronico </strong></td>
<td><label>
<input name="CorreoElectronico" type="text" id="CorreoElectronico" style="background:center"/>
</label></td></tr>
<tr><td><strong>Nivel de Acceso* </strong></td>
<td><label>
<input name="Nivel" type="text" id="Nivel" style="background:center"/>
</label></td></tr></table>
<span class="Estilo5">* Ingrese: <strong>1</strong> para Privilegios de Administrador<br />

```



```

        if (($strUsers == "") && false) {
            $isValid = true;
        }
    }
    return $isValid;
}
$MM_restrictGoTo = "indexerroradministrador.php";
if (!(isset($_SESSION['MM_Username'])) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup']))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_QUERY_STRING) && strlen($_QUERY_STRING) > 0)
        $MM_referrer .= "?" . $_QUERY_STRING;
    $MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
    header("Location: ". $MM_restrictGoTo);
    exit;
}
?>
<?php
mysql_select_db($database_conexion, $conexion);
$query_Recordset1 = "SELECT * FROM usuarios";
$Recordset1 = mysql_query($query_Recordset1, $conexion) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);
?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SEMOI_Aministrador_Registrar_Usuario_Confirmaci&oacute;n</title>
</head><body>
<div id="Layer2"><table width="100%" height="100%"><tr>

```

```

<td width="20%" valign="top"><div align="center" class="Estilo2"><br />
<span class="Estilo3">Seleccione la Opci&oacute;n de Administracion de Su Preferencia
</span></div>
<p align="center"><br />
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/button8.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/button8.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed> </object> </p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/ActualizarUsuario.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/ActualizarUsuario.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed> </object></p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="eliminar.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="eliminar.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock

```

```

waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed> </object></p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/confEv.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/confEv.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed> </object> </p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/button10.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/button10.swf" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed></object> </p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="BGCOLOR" value="#CBD1E9" />
    <param name="movie" value="animaciones/listarAlumnos.swf" />
    <param name="quality" value="high" />
    <embed src="animaciones/listarAlumnos.swf" width="120" height="28" quality="high"
pluginpage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" bgcolor="#CBD1E9" ></embed></object>

```

```

p></td> </table></div>
<div id="Layer4">
<?php
$conexion=mysql_connect("localhost","root")or die ("problemas en la conexion");
mysql_select_db("semoi", $conexion) or die ("problemas en la seleccion de la bdd");
$registros=mysql_query("select * from usuarios",$conexion) or die("problemas en el
select".mysql_error());
$usuario=$_REQUEST["Usuario"];
$band=0;
$contrasena=$_REQUEST["Contrasena"];
$nombreEst=ucwords ($_POST["NombreDeEstudiante"]);
$apellido=ucwords ($_POST["ApellidoDeEstudiante"]);
$correo=$_REQUEST['CorreoElectronico'];
$nivel=$_POST["Nivel"];
$x= strlen($nombreEst);
$x2=strlen($correo);
$email=explode('@',$correo);
$nombreCorreo=$email[0];
$extensionCorreo=$email[1];
$dominioCorreo=explode('.', $extensionCorreo);
$extension=$dominioCorreo[0];
$dominio=$dominioCorreo[1];
$clave='.';
if($nombreCorreo==" || $extension==" || $dominio=="||(!ereg("[a-zA-Z\@.\_]{5,35}$", $apellido)))
{
    echo '<script lenguaje="Javascript">location.href= "pagAdRegUsuarioErrorCorreo.php"
    </script>';
}
else if((!ereg("[a-zA-Z\ ]{3,20}$", $nombreEst))||(!ereg("[a-zA-Z\ ]{3,20}$", $apellido))
||strlen($usuario)<3||strlen($usuario)>20 || strlen($contrasena)<4||strlen($contrasena)>8 )
{
    echo '<script lenguaje="Javascript">location.href= "pagAdRegistrarUsuarioError2.php"
</script>';
}
else{

```



```

    <?php
    }
    else
    {
    echo '<script
language="Javascript">location.href="pagAdministradoRegUsuarioError.php"</script>';
    }
    }
mysql_close($conexion);
?>
</div><table width="100%"> <tr>
    <td><div align="center"></div></td> </tr>
    <tr> <td><div align="center"></div></td> </tr>
</table></body></html>
<?php
mysql_free_result($Recordset1);
?>

```

El resto de las páginas, a las cuales se hace referencia para el registro de usuario (pagAdministradoRegUsuarioError.php, pagAdRegistrarUsuarioError2.php, pagAdRegUsuarioErrorCorreo.php) contienen mensajes de error que se muestran al usuario en caso de que alguna entrada de datos sea incorrecta.

5.3.2.2 Integración del caso de uso Navegar Unidades

A continuación se muestra, parcialmente, la integración de la navegación de las unidades del software. Debido a que esta navegación es extensa, solo se presenta la interacción entre 2 lecciones, la visita al glosario, la ayuda, la evaluación y la actividad didáctica correspondiente. También, se mostrará la interfaz de cambio de

contraseña a la que el usuario tiene acceso desde cualquier unidad y/o lección de la aplicación.

Datos de Entrada: Específicamente, para la navegación de las unidades no se requieren datos de entrada, exceptuando, los correspondientes al login (usuario y contraseña), que le permiten acceder a esta sección del software.

Condiciones para la Navegación: Los usuarios o administradores deben estar correctamente “logueados” para acceder a la exploración de las unidades.

Procedimiento de prueba:

- Activar la interfaz de login del usuario o del administrador.
- En el caso del administrador, este debe ingresar a la página principal del administrador o cualquier otra pagina de la sesión del administrador, (Ver Fig. 5.15) la cual le proporcionará el acceso a la navegación de las unidades (pagina principal de unidades). En el caso del usuario, una vez que se introduzca el usuario y la contraseña correcta (proceso de login) se direcciona a la página principal de las unidades (Ver Fig. 5.16), que le permitirá la navegación de las mismas.
- Cada unidad posee un menú desplegable de botones que permitirán el acceso a las lecciones o unidades siguientes. En el caso, de que en la pagina principal se presione el link ‘Representación de la red’ (que se puede apreciar en la Fig 5.6), se redirecciona a la pagina correspondiente a esta lección. (Ver Fig.5.17)



Fig. 5.15. Pagina principal administrador (Explorar Unidades)

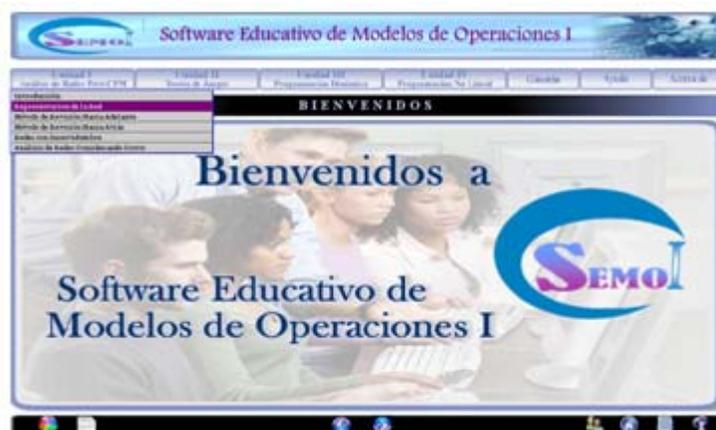


Fig. 5.16. Página principal unidades

Fig. 5.17. Interfaz de la lección 'Representación de la red' de la unidad I.

- Al presionar el botón ubicado en la parte inferior de la pantalla, correspondiente a ‘atrás’, se direcciona a la lección anterior, en este caso a la ‘Introducción’ de la unidad I, a la cual, se podrá tener acceso, igualmente, desde el menú desplegable. (Ver Fig. 5.18)



Fig. 5.18. Interfaz de la lección ‘Introducción’ de la unidad I

- Desde cualquier unidad o lección, a través de los botones ubicados en la parte inferior de la pantalla o en la parte superior de la misma, se puede acceder al glosario o a la ayuda del software (Fig.5.18). La interfaz del glosario se muestra en la Figura 5.19. En esta, el usuario debe elegir la opción de letra que desee, y, seguidamente aparecerán los conceptos asociados a ella. La interfaz de la ayuda se muestra en la Figura 5.20.



Fig. 5.19. Interfaz Principal de Glosario

- Igualmente, el usuario podrá cambiar su contraseña desde cualquier interfaz de unidad o lección del software. Para ello, se debe presionar el primer link ubicado en la parte inferior izquierda de la pantalla (Ver Fig. 5.18). La interfaz correspondiente al cambio de contraseña de usuario se muestra en la Fig. 5.21.



Fig. 5.20. Interfaz de Ayuda



Fig. 5.21. Interfaz de Cambio de Contraseña

- El acceso a la evaluación y a la actividad didáctica de la unidad en estudio, se realiza a través de los botones ubicados en la parte inferior derecha de la pantalla. Para la evaluación, el acceso estará restringido, solamente, a la última lección de dicha unidad (Ver Fig. 5.22). Es decir, si el usuario no se encuentra en la última lección se le denegará el acceso.

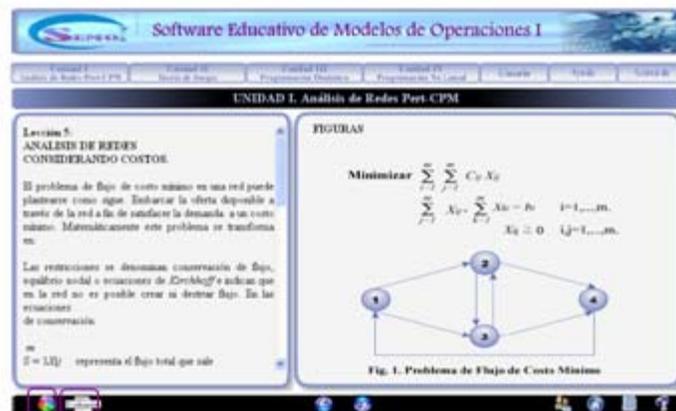


Fig. 5.22. Identificación de Links de evaluación y actividad didáctica

- Al presionar el primer link se dará acceso a la actividad didáctica correspondiente a la unidad 1. (Ver Fig. 5.23). Si se presiona el segundo link, se da acceso a la evaluación de esa unidad. (ver Fig. 5.24)



Fig. 5.23. Interfaz principal de la Actividad Didáctica (Unidad I)

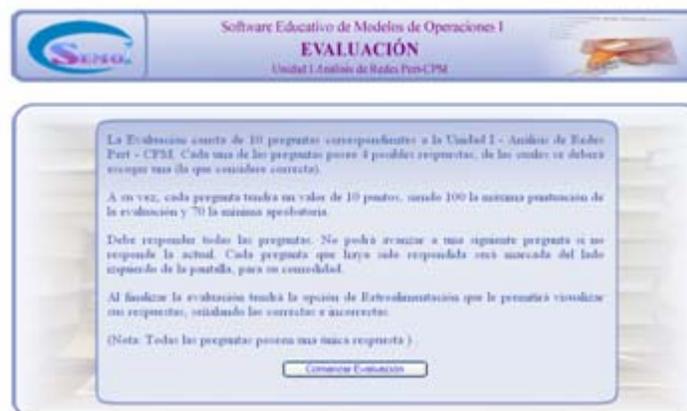


Fig. 5.24. Interfaz principal de la Evaluación (Unidad I)

Código Fuente del caso de uso Navegar contenido de Unidades

El código fuente correspondiente a las unidades fue presentado parcialmente en la fase de elaboración utilizando como ejemplo la lección 1 de la unidad 1 (Análisis de redes Pert-CPM). La diferencia entre este código y la del resto de las lecciones, se evidencia en el contenido correspondiente a cada lección. A continuación, se muestra

el código correspondiente a las interfaces mostradas durante las pruebas de unidad e integración de este caso de uso.

Código correspondiente a la Página Principal del Administrador del sistema (pagAdministrador.php):

```
<?php
if (!isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "1";
$MM_donotCheckaccess = "false";
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    $isValid = False;

    if (!empty($UserName)) {
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && false) {
            $isValid = true;
        }
    }
    return $isValid;
}
$MM_restrictGoTo = "indexerroradministrador.php";
if (!(isset($_SESSION['MM_Username']) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) {
```

```

$MM_qsChar = "?";
$MM_referrer = $_SERVER['PHP_SELF'];
if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
if (isset($QUERY_STRING) && strlen($QUERY_STRING) > 0)
$MM_referrer .= "?" . $QUERY_STRING;
$MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
header("Location: ". $MM_restrictGoTo);
exit;
}
?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Bienvenidos a la Sesión de Administrador</title>
<style type="text/css">
</style></head><body>
<div id="Layer2">
<table width="100%" height="100%"> <tr>
<td width="20%" valign="top"><div align="center" class="Estilo2"><br />
<span class="Estilo3">Seleccione la Opción de Administraci3n de Su Preferencia
</span></div>
<p align="center"><br />
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"codebase="http:// download.
macromedia. com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0" width="120" height="28">
<param name="movie" value="animaciones/button8.swf" />
<param name="quality" value="high" />
<param name="bgcolor" value="#CBD1E9" />
<embed src="animaciones/button8.swf" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed></object> </p>
<p align="center">

```

```

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/ActualizarUsuario.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed
        src="animaciones/ActualizarUsuario.swf"
        quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash"
        type="application/x-shockwave-flash"
        width="120"
        height="28"
bgcolor="#CBD1E9"></embed></object></p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="eliminar.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="eliminar.swf" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed></object></p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/confEv.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/confEv.swf" quality="high" pluginspage=
"http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash"
type="application/x-shockwave-flash" width="120" height="28" bgcolor="#CBD1E9"></embed>
</object> </p>
<p align="center">

```

```

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="movie" value="animaciones/explorarUnidades.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#CBD1E9" />
    <embed src="animaciones/explorarUnidades.swf" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" width="120" height="28"
bgcolor="#CBD1E9"></embed></object> </p>
<p align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"
width="120" height="28">
    <param name="BGCOLOR" value="#CBD1E9" />
    <param name="movie" value="animaciones/listarAlumnos.swf" />
    <param name="quality" value="high" />
    <embed src="animaciones/listarAlumnos.swf" width="120" height="28" quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock
waveFlash" type="application/x-shockwave-flash" bgcolor="#CBD1E9" ></embed>
</object></p></td></tr></table></div>
<table width="100%">
<tr> <td><div align="center"></div></td>
<td><div align="center"></div></td></tr></table>
</body></html>

```

Código correspondiente a la Página Principal Unidades (unidadpresentacion.php):

```
<?php session_start(); ?>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SEMOI_BIENVENIDOS</title>
<style type="text/css">
</style>
<script language="JavaScript">
<!--
function mmLoadMenus() {
    if (window.mm_menu_0318134401_0) return;
    window.mm_menu_0318134401_0 = new Menu("root",261,13,"Georgia, Times New Roman,
Times, serif",9,"#000000","#FFFFFF","#CCCCCC","#800080","left","middle",2,0,800,-
5,7,true,true,true,0,true,true);
    mm_menu_0318134401_0.addItem("Introducción","location='unidad1Introduccion.php'");
;
    mm_menu_0318134401_0.addItem("Representacion&nbsp;de&nbsp;la&nbsp;Red","locat
ion='unidad1leccion1.php'");
    mm_menu_0318134401_0.addItem("Método&nbsp;de&nbsp;Revisión&nbsp;Hacia&nbsp;
p;Adelante","location='unidad1leccion2.php'");
    mm_menu_0318134401_0.addItem("Método&nbsp;de&nbsp;Revisión&nbsp;Hacia&nbsp;
p;Atrás","location='unidad1leccion3.php'");
    mm_menu_0318134401_0.addItem("Redes&nbsp;con&nbsp;Incertidumbre","location='u
nidad1leccion4.php'");
    mm_menu_0318134401_0.addItem("Análisis&nbsp;de&nbsp;Redes&nbsp;Considerando
&nbsp;Costos","location='unidad1leccion5.php'");
    mm_menu_0318134401_0.hideOnMouseOut=true;
    mm_menu_0318134401_0.bgColor='#555555';
    mm_menu_0318134401_0.menuBorder=1;
    mm_menu_0318134401_0.menuLiteBgColor='#FFFFFF';
    mm_menu_0318134401_0.menuBorderBgColor='#0000FF';

```

```

window.mm_menu_0318142419_0 = new Menu("root",301,13,"Georgia, Times New Roman,
Times, serif",9,"#000000", "#FFFFFF", "#CCCCCC", "#800080", "left", "middle", 2,0,800,-
5,7,true,true,true,0,true,true);
mm_menu_0318142419_0.addItem("Introducción","location='unidad2Introduccion.php'");
mm_menu_0318142419_0.addItem("Formulación&nbsp;de&nbsp;Juegos&nbsp;de&nbsp;
;2&nbsp;Personas","location='unidad2leccion1.php'");
mm_menu_0318142419_0.addItem("Formulacion&nbsp;de&nbsp;Juegos&nbsp;Con&nbsp;
Esp&nbsp;Estrategias&nbsp;Mixtas","location='unidad2leccion2.php'");
mm_menu_0318142419_0.hideOnMouseOut=true;
mm_menu_0318142419_0.backgroundColor='#555555';
mm_menu_0318142419_0.menuBorder=1;
mm_menu_0318142419_0.menuLiteBgColor='#FFFFFF';
mm_menu_0318142419_0.menuBorderBgColor='#0000FF';
window.mm_menu_0318142839_0 = new Menu("root",279,13,"Georgia, Times New Roman,
Times, serif",9,"#000000", "#FFFFFF", "#CCCCCC", "#800080", "left", "middle", 2,0,800,-
5,7,true,true,true,0,true,true);
mm_menu_0318142839_0.addItem("Introducción","location='unidad3Introduccion.php'")
;
mm_menu_0318142839_0.addItem("Recurción&nbsp;en&nbsp;Reversa&nbsp;y&nbsp;e
n&nbsp;Avance","location='unidad3leccion1.php'");
mm_menu_0318142839_0.addItem("Modelo&nbsp;de&nbsp;la&nbsp;Ruta&nbsp;Más&
nbsp;Corta","location='unidad3leccion2.php'");
mm_menu_0318142839_0.addItem("Modelos&nbsp;de&nbsp;Tamaño&nbsp;de&nbsp;F
uerza&nbsp;de&nbsp;Trabajo","location='unidad3leccion3.php'");
mm_menu_0318142839_0.addItem("Aplicaciones&nbsp;de&nbsp;la&nbsp;Ppogramació
n&nbsp;Dinámica","location='unidad3leccion4.php'");
mm_menu_0318142839_0.hideOnMouseOut=true;
mm_menu_0318142839_0.backgroundColor='#555555';
mm_menu_0318142839_0.menuBorder=1;
mm_menu_0318142839_0.menuLiteBgColor='#FFFFFF';
mm_menu_0318142839_0.menuBorderBgColor='#0000FF';
window.mm_menu_0318143846_0 = new Menu("root",266,13,"Georgia, Times New Roman,
Times, serif",9,"#000000", "#FFFFFF", "#CCCCCC", "#800080", "left", "middle", 2,0,800,-
5,7,true,true,true,0,true,true);

```

```

    mm_menu_0318143846_0.addItem("Introducción","location='unidad4Introduccion.php");
    mm_menu_0318143846_0.addItem("Función&nbsp;Cónca&nbsp;y&nbsp;Convexa","location='unidad4leccion1.php");
    mm_menu_0318143846_0.addItem("Optimización&nbsp;No&nbsp;Restringida","location='unidad4leccion2.php");
    mm_menu_0318143846_0.addItem("Optimización&nbsp;Restringida","location='unidad4leccion3.php");
    mm_menu_0318143846_0.addItem("Método&nbsp;de&nbsp;Multiplicadores&nbsp;de&nbsp;LaGrange","location='unidad4leccion4.php");
    mm_menu_0318143846_0.hideOnMouseOut=true;
    mm_menu_0318143846_0.backgroundColor='#555555';
    mm_menu_0318143846_0.menuBorder=1;
    mm_menu_0318143846_0.menuLiteBgColor='#FFFFFF';
    mm_menu_0318143846_0.menuBorderBgColor='#0000FF';
    mm_menu_0318143846_0.writeMenus();
} // mmLoadMenus()
function MM_showMenu(menu, x, y, child, imgname) {
    if (!window.mmWroteMenu) return;
    MM_clearTimeout();
    if (menu) {
        var obj = FIND(imgname) || document.images[imgname] || document.links[imgname] || document.anchors[imgname];
        x = moveXbySlicePos (x, obj);
        y = moveYbySlicePos (y, obj);
    }
    if (document.layers) {
        if (menu) {
            var l = menu.menuLayer || menu;
            l.top = l.left = 1;
            hideActiveMenus();
            if (this.visibility) l = this;
            window.ActiveMenu = l;
        } else {
            var l = child;

```

```

    }
    if (!l) return;
    for (var i=0; i<l.layers.length; i++) {
        if (!l.layers[i].isHilite) l.layers[i].visibility = "inherit";
        if (l.layers[i].document.layers.length > 0) MM_showMenu(null, "relative", "relative",
l.layers[i]);
    }
    if (l.parentLayer) {
        if (x != "relative") l.parentLayer.left = x || window.pageX || 0;
        if (l.parentLayer.left + l.clip.width > window.innerWidth) l.parentLayer.left -=
(l.parentLayer.left + l.clip.width - window.innerWidth);
        if (y != "relative") l.parentLayer.top = y || window.pageY || 0;
        if (l.parentLayer.isContainer) {
            l.Menu.xOffset = window.pageXOffset;
            l.Menu.yOffset = window.pageYOffset;
            l.parentLayer.clip.width = window.ActiveMenu.clip.width +2;
            l.parentLayer.clip.height = window.ActiveMenu.clip.height +2;
            if (l.parentLayer.menuContainerBgColor && l.Menu.menuBgOpaque )
l.parentLayer.document.bgColor = l.parentLayer.menuContainerBgColor;
        }
    }
    l.visibility = "inherit";
    if (l.Menu) l.Menu.container.visibility = "inherit";
} else if (FIND("menuItem0")) {
    var l = menu.menuLayer || menu;
    hideActiveMenus();
    if (typeof(l) == "string") l = FIND(l);
    window.ActiveMenu = l;
    var s = l.style;
    s.visibility = "inherit";
    if (x != "relative") {
        s.pixelLeft = x || (window.pageX + document.body.scrollLeft) || 0;
        s.left = s.pixelLeft + 'px';
    }
}

```

```

        if (y != "relative") {
            s.pixelTop = y || (window.pageY + document.body.scrollTop) || 0;
            s.top = s.pixelTop + 'px';
        }
        l.Menu.xOffset = document.body.scrollLeft;
        l.Menu.yOffset = document.body.scrollTop;
    }
    if (menu) window.activeMenus[window.activeMenus.length] = l;
    MM_clearTimeout();
}
//-->
</script>
<script language="JavaScript" src="mm_menu.js"></script>
</head>
<body>
<div class="Estilo11" id="Layer2">
<div align="center" class="Estilo12">B I E N V E N I D O S</div>
</div>
<script language="JavaScript 1.2">mmLoadMenus();</script>
<table width="97%" height="130%" align="center">
<tr> <td height="78"><div align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0"
width="98%" height="73">
    <param name="movie" value="animaciones/encabezado.swf" />
    <param name="quality" value="high" />
    <embed
        src="animaciones/encabezado.swf"
        quality="high"
        pluginspage="http://www.macromedia.com/go/getflashplayer"
        type="application/x-shockwave-flash"
        width="98%" height="73"></embed> </object></div></td> </tr>
<tr><td width="95%" height="27"><div align="center"><a href="glosario.php" target="_blank"></a><a
href="ayuda.htm" target="_blank"></a><a href="pagAcercaDe.php" target="_blank"
onclick="window.open(this.href, this.target, 'width=520,height=420'); return false;"></a> </div></td>
</tr>
<tr><td width="95%" height="32" valign="bottom"><div align="center"></div></td></tr>
<tr><td width="95%" height="383" valign="bottom"><div align="center"></div></td></tr>
<tr><td height="19">
<div align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0
" width="98%" height="31">
<param name="movie" value="animaciones/BotInferiores.swf" />
<param name="quality" value="high" />
<embed src="animaciones/BotInferiores.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash"
width="98%" height="31"></embed> </object></div></td></tr></table>
</body>
</html>

```

Código correspondiente a la Página Principal de Glosario (glosario. php):

```
<?php session_start(); ?>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SEMOI_Glosario_Bienvenidos</title>
<style type="text/css">
</style></head><body>
<table width="99%" border="0"> <tr><td>
<div align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0"
width="97%" height="70">
<param name="movie" value="animaciones/encabezado.swf" />
<param name="quality" value="high" />
<embed src="animaciones/encabezado.swf" width="97%" height="70" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-
flash"></embed></object></div></td></tr>
<tr><td><div align="center"></div></td></tr>
<tr><td>
<div align="center"><a href="A.html"></a>
<a href="B.html"></a>
<a href="C.html"></a>
<a href="D.html"></a>
<a href="E.html"></a>
<a href="F.html"></a>
<a href="G.html"></a>
<a href="H.html"></a>
<a href="I.html"></a>
<a href="J.html"></a>
<a href="K.html"></a>
<a href="L.html"></a>

```

```

<a href="M.html"></a>
<a href="N.html"></a>
<a href="O.html"></a>
<a href="P.html"></a>
<a href="Q.html"></a>
<a href="R.html"></a>
<a href="S.html"></a>
<a href="T.html"></a>
<a href="U.html"></a>
<a href="V.html"></a>
<a href="W.html"></a>
<a href="X.html"></a>
<a href="Y.html"></a>
<a href="Z.html"></a>
</div></td> </tr>
<tr><td><div align="center"></div></td> </tr></table>
</body></html>

```

Código correspondiente a la página cambio de Contraseña (pagUsuario.php):

```

<?php
if (!isset($_SESSION)) {
    session_start();
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SEMOI_Usuario_Cambiar_Contrase&ntilde;a_Error</title>
<style type="text/css">
</style>

```

```

<script type="text/JavaScript">
<!--
function MM_findObj(n, d) { //
    var p,i,x; if(!d) d=document;
    if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);
    }
    if(!(x=d[n])&&d.all) x=d.all[n];
    for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
    if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_validateForm() {
    var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
    if (test!='R') { num = parseFloat(val);
        if (isNaN(val)) errors+='- '+nm+' Debe contener un numero.\n';
        if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);
        }
        else if (test.charAt(0) == 'R') errors += '- '+nm+' : No Debe ser NULO.Ingrese el valor
correspondiente.\n\n';
    }
    }
    if (errors) alert('ERROR:\n\n'+errors);
    document.MM_returnValue = (errors == "");
}
//-->
</script></head>
<body>
<div id="Layer7">
<form action="delete2.php" method="post"
onsubmit="MM_validateForm('NombreDeUsuario','R','ContrasenaActual','R','Nueva Contraseña'
,'R');return document.MM_returnValue">

```

```

<label><strong><br /> Nombre de Usuario</strong>
<input name="NombreDeUsuario" type="text" id="NombreDeUsuario" style="background:center" />
<br /> <br /></label>
<label><strong>Contraseña actual </strong>
<input name="ContrasenaActual" type="password" id="ContrasenaActual" style="
"background:center"/> <br /> <br /></label>
<label><strong>Nueva Contraseña </strong>
<input name="NuevaContrasena" type="password" id="NuevaContrasena" style="
"background:center"/></label>
<label><div align="center">
<input type="submit" name="Submit" value="Enviar" style="color:#0000FF" />
</div></label>
<div align="center"><br /> </div> </form></div>
<div id="Layer8">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0
" width="50%" height="29">
  <param name="movie" value="animaciones/usuarioayuda.swf" />
  <param name="quality" value="high" />
  <embed src="animaciones/usuarioayuda.swf" quality="high" pluginspage=
"http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash" width="50%"
height="29"></embed></object></div>
<table width="100%" height="95%"> <tr>
<td width="100%"><div align="center"></div></td> </tr>
</table></body></html>

```

Código correspondiente a la Página Principal de Actividad Didactica Unidad I (ActividadDidacticaUnidad1. php):

```

<?php session_start(); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SEMOI_UNIDAD_I_Actividad_Did&acute;tica</title>
<style type="text/css">
</style></head>
<body>
<div class="Estilo11" id="Layer2">
<div align="center" class="Estilo12">UNIDAD I. ACTIVIDAD
DID&acute;CTICA </div></div>
<div id="Layer8">
<table width="100%" border="0">
<tr><td><p align="justify" class="Estilo14"><br />
<span class="Estilo16">En esta actividad se probar&acute; tu conocimiento acerca
del an&acute;lisis y la representaci&oacute;n de Redes Pert - CPM. </span></p>
<p align="justify" class="Estilo17"> Debes responder verdadero o falso, para
cada pregunta que se presente. Al responder una pregunta pasar&acute;s
seguidamenete a la pr&oacute;xima y se indicar&acute; si esta fue correcta o
incorrecta.</p>
<p align="justify" class="Estilo13"><span class="Estilo15">El juego habr&acute;
terminado cuando contestes la mayor parte de las preguntas de forma correcta(Ganas)
o incorrecta(Pierdes).</span><br /> <br />

```

```

<span class="Estilo18">Para dar inicio a la actividad didactica presiona el siguiente
link </span> </p></td></tr>
<tr><td><div align="center"><a href="AD1.swf">JUGAR</a></div></td>
</tr></table></div>
<table width="97%%" height="130%" align="center">
<tr><td height="78"><div align="center">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase=
"http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,
0,19,0" width="98%" height="73">
<param name="movie" value="animaciones/encabezado.swf" />
<param name="quality" value="high" />
<embed src="animaciones/encabezado.swf" quality="high" pluginspage=
"http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-
flash" width="98%" height="73"></embed></object></div></td></tr>
<tr><td width="95%" height="32" valign="bottom"><div align="center"></div></td> </tr>
<tr><td width="95%" height="383" valign="middle"><div align="center"></div></td>
</tr></table>
</body></html>

```

**Código correspondiente a la Página Principal de Evaluación Unidad I
(Evaluacion1. php):**

```

<?php session_start();
$_SESSION["cont"]=0;
$_SESSION["bandRegreso"]=0;
$_SESSION["c"]=0;
$_SESSION["total1"]=1;
$_SESSION["total2"]=1;
$_SESSION["total3"]=1;

```

```

$_SESSION["total4"]=1;
$_SESSION["total5"]=1;
$_SESSION["total6"]=1;
$_SESSION["total7"]=1;
$_SESSION["total8"]=1;
$_SESSION["total9"]=1;
$_SESSION["total10"]=1;
$_SESSION["resp1"]='0';
$_SESSION["resp2"]='0';
$_SESSION["resp3"]='0';
$_SESSION["resp4"]='0';
$_SESSION["resp5"]='0';
$_SESSION["resp6"]='0';
$_SESSION["resp7"]='0';
$_SESSION["resp8"]='0';
$_SESSION["resp9"]='0';
$_SESSION["resp10"]='0';

```

```
?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>Evaluaci&oacute;n</title>
```

```
<style type="text/css">
```

```
</style></head>
```

```
<body>
```

```
<div id="Layer9">
```

```
<p align="justify" class="Estilo8"><br />
```

La Evaluación consta de 10 preguntas correspondientes a la Unidad I - Análisis de Redes Pert - CPM. Cada una de las preguntas posee 4 posibles respuestas, de las cuales se deberá escoger una (la que considere correcta).</p>

```
<p align="justify" class="Estilo12">A su vez, cada pregunta tendra un valor de 10 puntos, siendo 100 la máxima puntuación de la evaluación y 70 la mínima aprobatoria.
```

```
</p>
```

```
<p align="justify" class="Estilo12">Debe responder todas las preguntas. No podrá avanzar a una siguiente pregunta si no responde la actual. Cada pregunta que haya sido respondida será marcada del lado izquierdo de la pantalla, para su comodidad. </p>
```

```
<p align="justify" class="Estilo12">Al finalizar la evaluación tendrá la opción de Retroalimentación que le permitirá visualizar sus respuestas, señalando las correctas e incorrectas. </p>
```

```
<p align="justify" class="Estilo12">(Nota: Todas las preguntas poseen una única respuesta ) </p>
```

```
<p align="center" class="Estilo8">
```

```
<form id="form1" name="form1" method="post" action="Evaluacion11.php">
```

```
<div align="center">
```

```
<input type="submit" name="Submit" value="Comenzar Evaluación" style="color:#0000FF"
```

```
/> </p> </label> </div>
```

```
</form></div>
```

```
<table width="100%">
```

```
<tr><td align="center"></td> </tr>
```

```
<tr><td align="center"></td> </tr>
```

```
</table></body>
```

```
</html>
```

5.4 EVALUACIÓN DE LA FASE DE CONSTRUCCIÓN

En esta fase se implementó la totalidad del sistema y se realizaron las pruebas de unidad e integración necesarias para comprobar su correcto funcionamiento. Específicamente, en el flujo de trabajo de implementación se desarrolla el modelo de implementación del sistema, donde se diseña el diagrama de los componentes principales de la aplicación, los cuales corresponden al diseño de la arquitectura del software realizado.

Finalmente, se planificaron, diseñaron y ejecutaron las pruebas de unidad e integridad del sistema, dando como resultado la obtención del software en su primera versión operativa, la cual ofrece cubrir las expectativas generadas al comienzo de su desarrollo.

5.5 PLANIFICACIÓN DE LA FASE DE TRANSICIÓN

Una vez que el proyecto alcanza la capacidad operativa inicial se planifica la fase de transición. En la fase de transición, el software se pone a disposición de los usuarios finales, comprobando que el mismo cumpla con las necesidades planteadas. Se prueba la calidad del software y se completa su documentación (Se diseña la ayuda y los manuales de usuario).

CAPÍTULO VI: FASE DE TRANSICIÓN

6.1 INTRODUCCIÓN

Durante la fase de Construcción se obtuvo la primera versión operativa del software. Por lo tanto, en esta fase la labor se concentra en la preparación de la versión beta de la aplicación para que pueda ser utilizada por los usuarios finales. Esto incluye la planificación de la instalación de esta versión en el lugar elegido, y la preparación de la documentación requerida por el usuario para su instalación e implantación.

6.2 PLANIFICACIÓN DE LA INSTALACIÓN DE LA VERSIÓN BETA

Una vez se ha obtenido una versión operativa del software se debe elegir el lugar de la instalación y planificar la misma.

Para la instalación del Software educativo de modelos de operaciones I en el departamento de computación y sistemas de la Universidad de Oriente, deben habilitarse horas de laboratorio en la sala de micro “Rene Cabrera” o en la sala de “Navegación” del mencionado departamento, para los alumnos de esta materia. Si estas horas son habilitadas, se deben instalar el software en cada uno de los computadores existentes en la sala correspondiente.

A continuación se presentan los requisitos y el proceso de instalación del sistema:

- Instalar, en cada una de las computadoras de la sala habilitada, el servidor xamp, preferiblemente V.5.0. Luego, se deberá Importar la base de datos proporcionada, con el nombre de “Semoi”.

- Copiar todos los archivos correspondientes al software ('semoi') en la carpeta htdocs del servidor xamp que fue previamente instalado en el disco duro (C:\Archivos de programa\xampp\htdocs\).
- Instalar los plugins de FLASH, para permitir la ejecución de las animaciones. En su defecto, instalar directamente el programa Macromedia Flsah 8.
- Asegurarse de que cada computadora posea reproductor de música (sonidos).
- Ejecutar el programa desde el Navegador Internet Explorer (preferiblemente en su versión 8.0)

Una vez se instale correctamente el software en cada computador, el administrador del sistema (o el profesor) debe registrar a cada uno de los estudiantes de la sección correspondiente. De esta manera, cada usuario tendrá acceso a la aplicación. El usuario podrá cambiar su contraseña, si así lo desea. A partir de ese momento, el software podrá ser utilizado por la población estudiantil de la asignatura modelos de operaciones I, dictada en el departamento de computación y sistemas de la Universidad de Oriente.

6.3 DOCUMENTACIÓN

Luego de desarrollar por completo la primera versión operativa del software y decidir el sitio donde se instalará, se requiere de la documentación adecuada que sirva como guía de navegación para los usuarios, razón por la cual, se desarrolla el Manual de Usuarios del SEMOI (Ver Anexo C). En este, se explica de manera detallada los pasos a seguir para cada uno de los usuarios de la aplicación (administrador o usuario).

6.4 EVALUACIÓN DE LA FASE DE TRANSICIÓN

Al final de la fase de transición se obtiene un producto listo para utilizarse, debido a que se ha probado que cumple con los requerimientos, necesidades y objetivos planteados al inicio de este proyecto. Además, se proporcionan las herramientas y guías para que el usuario pueda acceder exitosamente a la aplicación.

En esta fase no fue necesario modificar significativamente algún componente de software, ya que su desarrollo, se finaliza durante la implementación de la fase anterior.

El software obtenido resulta una eficiente solución interactiva Web para proporcionar apoyo al proceso de enseñanza aprendizaje de la asignatura Modelos reoperaciones I.

CONCLUSIONES

- El sistema SEMOI es un software funcional, flexible, íntegro y robusto que cumple con los objetivos propuestos al inicio de su desarrollo.
- El Software Educativo de Modelos de Operaciones I (SEMOI), ofrece una alternativa para facilitar el proceso de enseñanza – aprendizaje de esta asignatura, en la Universidad de Oriente, mediante la aplicación de un sistema innovador, que da apoyo al modelo de educación tradicional aplicado a los estudiantes de esta casa de estudios.
- El desarrollo del SEMOI se basó en el Proceso Unificado de Desarrollo de Software, utilizando diagramas UML y las nociones básicas aportadas por Alvaro Galvis para el diseño de un Software educativo. Además, se utilizó la extensión Uml para aplicaciones Web (WAE), lo que permitió el desarrollo de un software orientado a la Web que sigue todas las pautas diseñadas para ello.
- En la fase de inicio del SEMOI, se puso en marcha el proyecto capturando parte de los requisitos del software. Esto permitió el desarrollo del modelo inicial de casos de usos, el cual fue refinado, posteriormente, en la fase de elaboración.
- En la fase de elaboración se estableció la línea base de la arquitectura, identificando el contenido que ofrece la aplicación y las estrategias necesarias para la transmisión efectiva del conocimiento.

- Durante el desarrollo del software se diseñaron micromundos interactivos con características atractivas, que promueven el interés del estudiante y facilitan el aprendizaje. De esta manera, se garantiza que el software cumple con su función educativa.
- Un software educativo requiere de un sistema que verifique si el aprendizaje, realmente, se está logrando. Por ello, se desarrollaron mecanismos de autoevaluación para cada una de las unidades de estudio del SEMOI. Estas evaluaciones permiten categorizar el conocimiento obtenido a través del estudio y definir si se han logrado los objetivos propuestos.
- Para la implementación del SEMOI, se emplearon las siguientes herramientas: el Lenguaje de Programación para Páginas Web (php), el lenguaje de creación de páginas Web (html), el manejador de Base de Datos (MySQL), el software para creación de animaciones (flash), y el asistente para el diseño y creación de imágenes (fireworks). El empleo de estas herramientas permitió la creación de un software educativo Web robusto y con características interesantes para los usuarios.
- El software cuenta con una base de datos en MySQL, que ofrece integridad y seguridad para los datos que se almacenan. Esto permitió el desarrollo de un sistema que ofrece acceso al software solo a los usuarios registrados. Adicionalmente, permitió el almacenamiento de preguntas y respuestas de las evaluaciones, a las cuales, se tiene acceso a través del sistema ofreciendo posibilidades de actualización de las mismas.

RECOMENDACIONES

- Se recomienda la puesta en práctica del sistema SEMOI. Esto permitirá que el software este al alcance de los estudiantes de la asignatura Modelos de Operaciones I del departamento de Computación y Sistemas de la Universidad de Oriente, para facilitar y completar el proceso educativo.
- Realizar mantenimiento y/o Actualización constante de la base de datos, de esta manera se garantiza que las preguntas y/o respuestas de la evaluación cumplan con los requisitos exigidos por el profesor y por el contenido programático de la asignatura.
- Se recomienda la implementación de otros casos de uso en el sistema, de esta manera, se proporcionará mayor funcionalidad a la aplicación. Entre estos casos de uso se pueden mencionar la inserción y eliminación de preguntas y/o respuestas en la evaluación.
- Realizar pruebas con los usuarios finales del SEMOI y desarrollar una nueva versión, si es necesario. De esta manera, se podría examinar el código fuente de la aplicación y de ser posible optimizar los componentes.
- Proporcionar un entrenamiento adecuado para los usuarios y administradores del sistema.

- Estudiar la posibilidad de colocar el software a disposición de los estudiantes de la Universidad de Oriente, a través del portal Aula Virtual perteneciente al sitio Web www.anz.udo.edu.ve.
- Se recomienda promover y dar apoyo al trabajo multidisciplinario en tesis de grado en la Universidad de Oriente. En este caso, se logró la ejecución satisfactoria del Software Educativo de Modelos de Operaciones I, con la participación de una estudiante de Ingeniería en Computación y una estudiante de Ingeniería de Sistemas.

BIBLIOGRAFÍA

- [1] Abbondanza J., Rodríguez A., (2004), “Desarrollo de un Software educativo Interactivo sobre los fundamentos de Inteligencia Artificial utilizando Tecnología World Wide Web”. Tesis de Grado de Ingeniería en Computación, Universidad de Oriente, Barcelona, Venezuela.
- [2] Veracierta, G., Torres, P., (2005), “Desarrollo de un Software Educativo para Facilitar la Enseñanza y Aprendizaje de la Asignatura Análisis y Diseño de Algoritmos de las Unidades I, II, III y V; Dictada en el Departamento de Computación y Sistemas de la Universidad de Oriente Núcleo Anzoátegui”. Tesis de Grado de Ingeniería en Computación, Universidad de Oriente, Barcelona, Venezuela.
- [3] Rodríguez, R., (2005), “Desarrollo de un Software Educativo Interactivo que Sirva de Apoyo en el Proceso de Enseñanza – Aprendizaje de la Asignatura Matemática De Octavo Grado Impartida en una Unidad Educativa”. Tesis de Grado de Ingeniería en Computación, Universidad de Oriente, Barcelona, Venezuela.
- [4] Briceño, B., Gil, Y., (2006), “Desarrollo de un Software Educativo como Herramienta de Apoyo al Proceso de Enseñanza Aprendizaje de la Cátedra Ciencias Biológicas de Noveno Grado de Educación Básica, Utilizando la Tecnología Multimedia y World Wide Web”. Tesis de Grado de Ingeniería en Computación, Universidad de Oriente, Barcelona, Venezuela.
- [5] Pereira M., (2006), “Desarrollo de un Software Instruccional para la Enseñanza – Aprendizaje de la Asignatura Programación I que se Imparte en el Departamento de Computación y Sistemas de la Universidad de Oriente Núcleo Anzoátegui”. Tesis de Grado de Ingeniería en Computación, Universidad de Oriente, Barcelona, Venezuela.

- [6] Buratto, C., Canaparo, A., Laborde, A., Minelli A., (2004), “La Informática como Recurso Pedagógico-Didáctico en la Educación”, tomado de: <http://www.monografias.com/trabajos10/recped/recped.shtml#pea>. En fecha: Mayo, 2008.
- [7] Salcedo, P., (2002), “Ingeniería de Software Educativo, Teorías y Metodologías que la Sustentan”, tomado de: <http://www.inf.udec.cl/revista/ediciones/edicion6/isetm.PDF>. En fecha: Mayo, 2008.
- [8] Gomez, R., Galvis, A., Mariño, O., (2005), “Ingeniería de Software Educativo con Modelaje Orientado por Objetos: un Medio para Desarrollar Micromundos Interactivos”, tomado de: http://www.redacademica.edu.co/redacad/export/REDACADEMICA/directivos/proyectos_pedagogicos/micromundos/El_Proyecto_Micromundos/Documentos/pdf/ISEOO.pdf. En fecha: Mayo, 2008.
- [9] Inacmes, Currículo Universitario, Unidad de Desarrollo Virtual de la Universidad de Caldas, (2007), “Micromundos Interactivos para el Aprendizaje en una Escuela Rural”, tomado de: <http://fimunevar.files.wordpress.com/2007/03/ponencia-micromundos-semillero-edumatica.doc>. En fecha: Mayo, 2008.
- [10] Dr. Marqués Graells, P., (2003), “Estructura Básica De Los Materiales Multimedia”, tomado de: <http://dewey.uab.es/PMARQUES/estructu.htm>. En fecha: Mayo, 2008.
- [11] Booch, G., Jacobson, I., Rumbaugh, J., (2000), “El Proceso Unificado de Desarrollo de Software”. 1era Edición. Editorial Pearson Addison Wesley. España.
- [12] Aquilino, A., Fuente, J., Cueva, M., (2006), “Justificación Del Uso Del Proceso Unificado De Desarrollo De Software Como Modelo”, tomado de: <https://forja.rediris.es/docman/view.php/227/369/Proceso%20Unificado.Pdf>. En fecha: Junio, 2008.

- [13] Salinas, P., Histchfeld, N., (2005), "Tutorial de UML", tomado de: www.dcc.uchile.cl/~psalinas/uml/introduccion.html. En fecha: Junio, 2008.
- [14] García, Z., (2006), "Proceso Unificado de Desarrollo de Software Usando UML". Programa de Formación Continua, Universidad de Oriente.
- [15] Perez, M., (2003), "UML", tomado de: <http://www.monografias.com/trabajos7/bada.shtml>. En fecha: Junio, 2008.
- [16] Ijelchuk, C., (2003), "Lenguaje de Modelado Unificado", tomado de: <http://www.clikear.com/manuales/uml>. En fecha: Junio, 2008.
- [17] Fernández, A., (2001), "Diagrama de Componentes", tomado de: <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>. En fecha: Junio, 2008.
- [18] Fernández, A., (2001), "Diagrama de Implementación", tomado de: <http://www-gris.det.uvigo.es/~avilas/UML/node48.html>. En fecha: Junio, 2008.
- [19] Fernández, A., (2001), "Diagrama de Despliegue", tomado de: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>. En fecha: Junio, 2008.
- [20] Fernández, A., (2001), "Diagrama de Estado", tomado de: <http://www-gris.det.uvigo.es/~avilas/UML/node45.html>. En fecha: Junio, 2008.
- [21] Fernández, A., (2001), "Diagrama de Actividades", tomado de: <http://www-gris.det.uvigo.es/~avilas/UML/node46.html>. En fecha: Junio, 2008.
- [22] Fernández, A., (2001), "Diagrama de Objetos", tomado de: <http://www-gris.det.uvigo.es/~avilas/UML/node39.html>. En fecha: Junio, 2008.
- [23] Ocaño, J., López, M. (2002), "Introducción a la Ingeniería WEB Basada en UML", tomado de: <http://www.cs.buap.mx/~cuartocongreso/webs/apdf/A16.pdf>. En fecha: Junio, 2008.

- [24] Rendón Gallón, Á., (2001), “Modelado de aplicaciones en Internet”, tomado de: <ftp://jano.unicauca.edu.co/cursos/Especializacion/ApliServicios/docs/ModeladoInternet.pdf>. En fecha: Junio, 2008.
- [25] Escudero, S., (2002), “Macromedia Flash MX”. 1era Edición. Editorial McGraw Hill. España.
- [26] Martín, P., (2006), “HTML”, tomado de: <http://www.asptutor.com/zip/cbhtml.pdf>. En fecha: Junio, 2008.
- [27] Casares, C., (2004), “Tutorial de SQL”, tomado de: <http://www.maestrosdelweb.com/editorial/tutsq1/>. En fecha: Junio, 2008.
- [28] Gracia, J., (2006), “Conceptos Básicos de PHP”, tomado de: <http://www.webestilo.com/php/php00.phtml>. En fecha: Junio, 2008.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

TÍTULO	DESARROLLO DE UN SOFTWARE EDUCATIVO PARA FACILITAR EL PROCESO ENSEÑANZA-APRENDIZAJE DE LA ASIGNATURA MODELOS DE OPERACIONES I, DICTADA EN EL DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS DE LA UNIVERSIDAD DE ORIENTE
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
SALAZAR R., MARIA A.	CVLAC: 17.236.758 E MAIL: aiaemrl@hotmail.com
CADENILLAS E., REYNA I.	CVLAC: 17.658.853 E MAIL: cadenillasri@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Software

Educativo

Modelos de Operaciones I

Web

Proceso enseñanza - aprendizaje

Computacion y Sistemas

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

ÁREA	SUBÁREA
INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN
	INGENIERÍA DE SISTEMAS

RESUMEN (ABSTRACT):

El trabajo de grado presentado, consiste en el desarrollo de un software educativo que dará apoyo al sistema educativo empleado en el departamento de computación y sistemas de la Universidad de Oriente, facilitando el proceso de enseñanza-aprendizaje de la asignatura Modelos de Operaciones I, dictada como cátedra reglamentaria para ingeniería de sistemas y como electiva técnica para ingeniería en computación. Con la realización del software educativo de Modelos de Operaciones I (SEMOI) se persigue el desarrollo de un proyecto completo que ofrezca una alternativa tecnológica e interactiva para el sistema de educación tradicional y que, a su vez, permita reforzar conocimientos y facilite la labor docente a través del uso de tecnología multimedia y la World Wide Web. En el SEMOI se expondrá de manera didáctica, clara y detallada cada uno de los temas que forman parte del contenido de la asignatura, utilizando diversas herramientas audiovisuales como imágenes estáticas y dinámicas, juegos y animaciones, además, se realizan evaluaciones al final de cada capítulo que permitirán, tanto a profesores como a estudiantes, determinar si el aprendizaje se está llevando a cabo de una manera efectiva. Para el diseño y construcción del software educativo se utiliza Macromedia Dreamweaver 8, para la implementación de la aplicación en una página HTML y/o PHP que será mostrada en Internet; Macromedia Flash 8; y MySQL para la implementación de la base de datos que sirve de apoyo para la ejecución de las evaluaciones y soporte de información de usuarios.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
	ROL	CA	AS X	TU	JU
Garcia L., Zulirais D.	CVLAC:	10.299.576			
	E_MAIL	zzzuliii@hotmail.com			
	E_MAIL				
Saettone, Mónica.	ROL	CA	AS	TU	JU X
	CVLAC:	10.948.010			
	E_MAIL	monicasaettone@gmail.com			
	E_MAIL				
Dorta, Pedro.	ROL	CA	AS	TU	JU X
	CVLAC:	12.914.617			
	E_MAIL	dortap@hotmail.com			
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	07	28
AÑO	MES	DÍA

LENGUAJE. SPA

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.SofwareEducativoModelosI.doc	application/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J
K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u
v w x y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación / Ingeniero de Sistemas

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente, Núcleo de Anzoátegui

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

DERECHOS

Los trabajos de grado son de exclusiva propiedad de la
Universidad y sólo podrán ser utilizados para otros fines con el
conocimiento del Consejo de Núcleo respectivo, quién lo
participará al Consejo Universitario

Salazar R., Maria A.

AUTOR

Cadenillas E., Reyna I.

AUTOR

Garcia Zulirais

TUTOR

Saettone Mónica

JURADO

Dorta Pedro

JURADO

POR LA SUBCOMISION DE TESIS