



**Universidad de las Ciencias Informáticas
Facultad 4**

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

**ANÁLISIS Y DISEÑO DEL LABORATORIO VIRTUAL DE BIOLOGÍA EN LA
PLATAFORMA EDUCATIVA ZERA**

AUTORA:

Ediana González Machado

TUTOR:

Ing. Yosnel Herrera Martínez

La Habana, 2011

“Año 53 de la Revolución”



“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible”.

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del trabajo “Análisis y Diseño del laboratorio virtual de Biología en la Plataforma Educativa ZERA” y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Ediana González Machado

Ing. Yosnel Herrera Martínez

Tutor:

Yosnel Herrera Martínez: Profesor Instructor e Ingeniero en Ciencias Informáticas. Aspirante a Máster en Informática Aplicada en el área del modelado del Software Educativo. Profesor de las asignaturas de pregrado: Introducción a la Informática, Práctica Profesional, Introducción a las Tecnologías para la Formación y Diseño Instruccional. Investigador en el área de Software Educativo desde hace alrededor de 5 años, trabajando como: analista, diseñador y programador de este tipo de aplicaciones. Ha publicado artículos tanto en solitario como en colaboración con otros autores y participado en eventos científicos nacionales e internacionales sobre este ámbito.

Correo: yherrera@uci.cu

AGRADECIMIENTOS

A mis padres Edith y Pedro, por apoyarme siempre en todo y estar ahí dándome aliento en los momentos más difíciles de la carrera. Les agradezco por cada segundo de comprensión y paciencia que han tenido conmigo durante toda mi vida, por darme siempre fuerzas para seguir adelante y demostrarme que yo sí podía llegar hasta aquí.

A mis abuelos Ibrahim y China por ser los mejores abuelos del mundo, por brindarme todo su apoyo a lo largo de toda la carrera y estar siempre pendientes de mí.

A Yasniel por apoyarme siempre y brindarme toda su ayuda incondicional en las buenas y en las malas. Mil gracias por estar siempre pendiente de mí.

A mi tutor Yosnel por su apoyo, ayuda, dedicación y entrega incondicional que me brindó en este período para culminar satisfactoriamente la realización de este trabajo de diploma, sus consejos día a día contribuyeron inmensamente a mi formación como futura profesional.

Al profesor Dosagües por su apoyo y ayuda incondicional en todo momento, eternamente agradecida profe, por sus sabios consejos desde el comienzo y por demostrarme que si se podía alcanzar la meta.

A toda mi familia en general, en especial mi tía Elva y a mi prima Gredys por apoyarme, ayudarme y confiar en mí siempre.

A Dixán por ser como eres, tan especial, por el apoyo y los sabios consejos que me distes en los momentos finales, aunque no puedas estar físicamente a mi lado yo sé que estás muy orgulloso de mí.

A mis padres adoptivos de Venezuela, Maribel y Julio, a tí Mary por ser mi segunda mamá y por apoyarme tanto y darme aliento cuando más lo necesitaba.

Al profesor Marcel por su gran aporte en la confección y recopilación de la información y contenidos requeridos para el desarrollo de este trabajo.

A todos mis compañeros del aula ,de cuarto y amigos de estos cinco años de la carrera con los cuales he compartido momentos inolvidables, en especial a Lizzi, Lili, La Yure, Irena, La Yola, Yordi, Raidel, Yerandy, Adrián y Migue.

A nuestra Revolución y al compañero Fidel por haber confiado en todos nosotros y permitirnos estudiar en un centro de altos estudios como es hoy la Universidad de las Ciencias Informáticas.

Gracias a los que por olvido no mencioné y siempre preguntaban al verme: y la tesis ¿cómo va? Gracias a todas y todos por todo.

DEDICATORIA

A mis padres, quienes me han brindado amor incondicional y siempre me han impulsado a superarme profesionalmente, ofreciéndome aliento constante para lograrlo. Ustedes constituyen mi guía, mi fuerza y mi razón de ser. A mi hermanito Pedritín por todo el cariño, y por constituir uno de mis principales tesoros en este mundo. Aquí está el sueño de ustedes...

RESUMEN

Las Tecnologías Informáticas son usadas a nivel mundial, en el desarrollo de software para la educación en todos sus niveles dentro del proceso docente educativo. Existen diversas plataformas educativas para la gestión del aprendizaje, algunas de ellas, no se adaptan a determinadas necesidades del modelo pedagógico cubano, como son la no existencia de los hiperentornos de aprendizaje.

La investigación que se presenta tiene por objetivo el desarrollo de un análisis y diseño, para la implementación de un laboratorio virtual de Biología, para la Plataforma de Gestión del Aprendizaje ZERA, la cual permite agilizar procesos, reducir costos y contribuir al proceso de aprendizaje, así como la adquisición de conocimientos y habilidades intelectuales. El estudio se realiza a partir de métodos científicos que posibilitan el desarrollo del objeto de investigación. Se describen las metodologías de desarrollo y herramientas seleccionadas para el desarrollo del producto. La investigación refiere el desarrollo de la solución propuesta, para lo que se identifica el modelo conceptual del problema, se determinan los requerimientos funcionales y no funcionales del sistema. Se identifican los entornos de presentación, diagramas de estructura de presentación y navegación basados en la utilización del lenguaje de modelado ApEM-L. El resultado de la investigación, muestra un análisis y un diseño que permite la obtención de una documentación ingenieril para el desarrollo de un Laboratorio Virtual de Biología.

Se presentan recomendaciones que permiten perfeccionar la solución propuesta, así como una exposición escrita, de la bibliografía empleada.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	5
1.1-Conceptos asociados al dominio del problema	5
1.1.1-Software Educativo.....	5
1.1.2-Proceso de enseñanza-aprendizaje	7
1.1.3-Hiperentorno de Aprendizaje	7
1.1.4-Plataformas Educativas.....	8
1.1.5-Laboratorios Virtuales	9
1.2-Análisis de soluciones existentes en el desarrollo de Laboratorios Virtuales.....	11
1.3-Metodologías de desarrollo	14
1.3.1-Metodología eXtreme Programing (XP).....	15
1.3.2-Proceso Unificado de Desarrollo de Software (RUP).....	15
1.3.3-Proceso para el desarrollo de proyectos de SWE en la UCI.	17
1.4-Arquitectura y Patrones	18
1.4.1-Patrón arquitectónico Modelo Vista Controlador	18
1.4.2-Patrón arquitectónico Modelo Vista Controlador Entidad (MVC-E)	19
1.4.3-Patrones de diseño.....	20
1.5-Lenguajes de modelado	21
1.5.1-Lenguaje Unificado de Modelado (UML).	22
1.5.2-Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L)	23
1.5.3-El Lenguaje de Modelado orientado a objeto para Aplicaciones Educativas y Multimedia (ApEM-L)	25
1.6-Herramientas de modelado	27
1.6.1-Argo UML.....	28
1.6.2-Visual Paradigm	28
1.7-Lenguaje de Desarrollo	29
1.7.1-ActionScript 3.0.....	29
1.8-Framework de Desarrollo	29
1.8- Flex SDK 4.0.....	29
1.9-Propuesta de solución técnica	30

CAPÍTULO II: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	32
2.1- Modelado del Dominio.....	32
2.1.1-Análisis de los conceptos del dominio.	32
2.1.2-Diagrama de clases del Modelo de Dominio.	34
2.2-Descripción del sistema propuesto.....	36
2.3-Descripción de la funcionalidad.....	36
2.3.1-Requerimientos Funcionales.....	37
2.3.2-Requerimientos No Funcionales.	38
2.4-Descripción de los actores que interactúan con el sistema.	39
2.5-Vista de Gestión del Modelo y sus Subsistemas.....	40
2.6-Descripción textual de los Entornos de Presentación.	42
2.7-Diagramas de Estructura de Presentación.....	51
2.8-Diagramas de Estructura de Navegación.....	53
CAPÍTULO III: DISEÑO DE LA SOLUCIÓN PROPUESTA	55
3.1- Patrón Arquitectónico y Patrones de Diseño Utilizados.....	55
3.2-Modelo del Diseño.....	57
3.2.1- Diagrama de Clases.....	58
3.2.2-Diagramas de Secuencia.....	60
3.3-Diseño de la Base de Datos.....	62
3.3.1-Descripción de las tablas.....	62
CONCLUSIONES GENERALES	63
RECOMENDACIONES	64
REFERENCIAS BIBLIOGRÁFICAS	65
BIBLIOGRAFÍA	68

INTRODUCCIÓN

Las tecnologías de la información y las comunicaciones (TIC), como parte de la llamada Revolución Científica Tecnológica, constituyen una ayuda en todos los procesos de la sociedad. En la esfera de la educación las TIC constituyen herramientas de alto valor, que contribuyen al proceso de enseñanza y aprendizaje, entre los que se destacan los sitios Web, entornos virtuales, plataformas, entre otros.

En la actualidad, algunas de las plataformas educativas desarrolladas para la gestión del aprendizaje virtual, no se adecuan a necesidades del modelo pedagógico cubano, como son por ejemplo, aquellas que no contemplan en su estructura los hiperentornos de aprendizaje. Su empleo amplía los servicios y proporciona variadas herramientas informáticas, que apoyan y facilitan el proceso de aprendizaje.

En la Facultad cuatro, perteneciente a la Universidad de las Ciencias Informáticas, se desarrolla una plataforma de contenido educativo nombrada ZERA, la que contiene normas y principios establecidos por pedagogos y especialistas del Ministerio de Educación de Cuba (MINED).

Con el objetivo de incrementar los servicios, para la plataforma ZERA es de interés, el desarrollo de nuevas prestaciones, que amplíen y respondan a demandas del proceso de enseñanza-aprendizaje de diferentes instituciones educacionales. La realización de prácticas de laboratorio de la disciplina de Biología, son necesidades que imprimen un estudio para su desarrollo.

En esta disciplina, existen debilidades en la presentación de sus contenidos a través de la plataforma. Entre sus limitantes se encuentra, que los profesores necesitan desarrollar competencias y habilidades en sus educandos y no cuentan con la posibilidad de mostrar, (a todos los estudiantes a su vez), los procesos que forman parte de experimentos y otros ejercicios docentes, por falta de medios y recursos, que tributen a la orientación del aprendizaje y al desarrollo de competencias; problemática que se ha agudizado en los últimos cursos, al incrementarse la carencia de los recursos, lo que impide una adecuada preparación.

Teniendo en cuenta la situación presentada, el **problema a resolver** de la presente investigación se centra en: ¿cómo realizar la simulación de prácticas de laboratorio de Biología, sobre la base de los hiperentornos de aprendizaje, que formen parte de la plataforma de gestión del aprendizaje ZERA?

Para darle solución al problema descrito se planteó como **objetivo general** de la investigación: Desarrollar el análisis y el diseño de un laboratorio virtual de Biología para la plataforma de gestión del aprendizaje ZERA que permita obtener una documentación ingenieril para su implementación.

El **objeto de estudio** que será investigado, es el proceso de desarrollo de los laboratorios virtuales, y como **campo de acción** enmarcando los resultados científicos en el área de los laboratorios virtuales de Biología.

Para guiar la investigación se plantea la siguiente **idea a defender**: Si se realiza el análisis y el diseño para el desarrollo de un laboratorio virtual para la disciplina de Biología, se podrán realizar simulaciones de prácticas de laboratorio de Biología, sobre la base de los hiperentornos de aprendizaje contemplados en la plataforma de gestión del aprendizaje ZERA.

Los **objetivos específicos** para el desarrollo de la presente investigación son:

- ✓ Realizar el estudio del estado del arte de la investigación.
- ✓ Definir los requerimientos funcionalidades y no funcionales para el desarrollo del análisis y diseño del laboratorio virtual de Biología.
- ✓ Analizar y diseñar una propuesta de solución del laboratorio virtual de Biología para la plataforma ZERA.

Las tareas que dan cumplimiento a los objetivos descritos son:

- ✓ Realización del análisis bibliográfico para conformar la base teórica y metodológica de la investigación.
- ✓ Investigación de las tendencias y las tecnologías actuales.
- ✓ Investigación del estado del arte de soluciones similares de laboratorios virtuales destinados para la gestión del aprendizaje.
- ✓ Realización de un Plan de Captura de Requerimientos.
- ✓ Descripción de las funcionalidades del sistema.
- ✓ Realización del análisis y diseño de una propuesta de solución.

Los métodos que soportan la presente investigación se basan en la combinación dialéctica de métodos Empíricos y Teóricos. Como empíricos, se emplean **la observación** para recopilar datos relevantes durante el proceso de investigación de los laboratorios virtuales. Este método posibilitará investigar con claridad los aspectos involucrados y definir los conceptos fundamentales para llevar a cabo la búsqueda

de información. El método **Análisis Documental**, se emplea para recopilar toda la información respecto al módulo Laboratorio del subsistema Simuladores de la Plataforma ZERA.

Entre los métodos de investigación teóricos, se emplea el método **histórico-lógico**, para conocer la evolución y desarrollo de los laboratorios virtuales. Este método proporciona resultados lógicos y concretos sobre los procesos que soporten el análisis y el diseño de la investigación. El **método analítico-sintético** se emplea para el análisis de la teoría y extracción de los principales conceptos a incluir en el marco teórico, además para el estudio y análisis de la información, lo que permite obtener los elementos principales relacionados con la plataforma de gestión de aprendizaje.

La tesis se **estructura** de la siguiente forma: presenta una introducción, tres capítulos, las conclusiones, con sus respectivas recomendaciones, además de darle tratamiento a la bibliografía consultada para el completamiento del amplio marco teórico y el conjunto de los anexos de la misma.

Los capítulos que se abordan son los siguientes:

- ✓ **Capítulo No. Uno:** *Fundamentación teórica*. En el presente capítulo, se expone el estado del arte y los conceptos vinculados al objeto de estudio, con el fin de lograr mejor comprensión del problema de la investigación. Se describe además, metodologías de desarrollo, y sistemas para realizar el análisis y diseño de la aplicación, así como las tendencias y tecnologías actuales sobre las que se basa. Finalmente se da a conocer la selección de los elementos que van a permitir el desarrollo del producto informático.
- ✓ **Capítulo No. Dos:** *Descripción de la solución propuesta*. En el presente capítulo, se muestra el modelado del contexto del problema a través del Modelo del Dominio. Se especifican los requerimientos funcionales y no funcionales que responderán a la futura implementación del sistema. Además se muestra la Vista de Gestión del Modelo con los subsistemas y sus relaciones, sobre la base del principio de organización del sistema definido por el Lenguaje de Modelado orientado a objeto para Aplicaciones Educativa y Multimedia (ApEM-L) en su versión 1.5 (1), el cual involucra a los “*Entornos de Presentación*” como artefacto principal para el modelado del análisis. Finalmente se presenta una muestra de las especificaciones textuales de algunos de los Entornos de Presentación.
- ✓ **Capítulo No. Tres:** *Diseño de la solución propuesta*. En el presente capítulo, se incluye la

descripción de los estilos arquitectónicos y patrones utilizados. Define el modelo de diseño del sistema. Muestra los diagramas de clases, de secuencias y se presenta el Modelo-Entidad-Relación de las clases persistentes del sistema a construir.



CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se plantean los principales conceptos relacionados con el objeto de estudio y que resultan vitales para entender la solución propuesta. Se aborda sobre el análisis del estado del arte de los laboratorios virtuales de Biología aplicado en software educativo, partiendo del estudio de los diferentes conceptos fundamentales para hacer la investigación. Además, se realiza un análisis de las metodologías y herramientas existentes para el desarrollo del software.

1.1- Conceptos asociados al dominio del problema

Toda investigación lleva implícito el desarrollo y selección de los principales conceptos que la soportan. En la presente, se desarrollan los conceptos de Software Educativo, Laboratorios Virtuales, Plataformas Educativas, Hiperentornos de Aprendizaje, Proceso de Enseñanza y Aprendizaje. A continuación se desarrollan los mismos.

1.1.1-Software Educativo

Del resultado de la vinculación de la Enseñanza con las nuevas tecnologías de la información y las comunicaciones (TIC) nace la parte de la ciencia dedicada al proceso de selección, elaboración, diseño y explotación de los recursos informáticos dirigidos a la gestión docente: la Informática Educativa y con ella el resultado científico más importante en el contexto educacional, el Software Educativo. Según el investigador chileno Jaime Sánchez (1999) *“el Software Educativo se define como cualquier programa computacional cuyas características estructurales sirvan de apoyo al proceso de enseñar, aprender y administrar. Este se caracteriza por ser altamente interactivo, a partir del empleo de recursos multimedia, como videos, sonidos, fotografías, diccionarios especializados, explicaciones de experimentados profesores, ejercicios y juegos instructivos que apoyan las funciones de evaluación y diagnóstico”* (2).

Actualmente existen muchas otras definiciones en la literatura por diferentes autores tanto internacionales como cubanos. Aunque no hay una definición exacta, todos coinciden que son sistemas informáticos que

responden al proceso de enseñanza-aprendizaje y se emplean en diferentes ramas del quehacer cultural, donde la educación juega un factor importante.

Una característica principal en la actualidad es que muchos de estos sistemas emplean diferentes tecnologías como la multimedia e hipermedia. *“Todas las formas de software educativos han sido absorbidas por esta tecnología, lo cual no es pura casualidad, sino el resultado de un proceso histórico que ha pretendido combinar los diferentes métodos para transmitir la información, en esperanza de una mayor calidad del propio proceso de adquisición de conocimientos.”* (3). Sin embargo la característica más relevante de los últimos tiempos es la tendencia de desarrollar este tipo de sistemas a entornos web.

Al mismo tiempo en el ámbito internacional los SWE se destacan por presentar una heterogeneidad de clasificaciones según su estructura y atendiendo a su función didáctica en el proceso de enseñanza-aprendizaje. Algunas de estas clasificaciones son: (4)

- ✓ Sistemas Tutoriales. Sistema basado en el diálogo con el estudiante, adecuado para presentar información objetiva, tiene en cuenta las características del alumno, siguiendo una estrategia pedagógica para la transmisión de conocimientos.
- ✓ Sistemas Entrenadores. Se parte de que los estudiantes cuentan con los conceptos y destrezas que van a practicar, por lo que su propósito es contribuir al desarrollo de una determinada habilidad, intelectual, manual o motora, profundizando en las dos fases finales del aprendizaje: aplicación y retroalimentación.
- ✓ Libros electrónicos. Su objetivo es presentar información al estudiante a partir del uso de sonido, texto, gráficos, animaciones y videos, pero con un nivel de interactividad y motivación que le facilite las acciones que realiza.
- ✓ Simuladores. Su objetivo es apoyar el proceso de enseñanza aprendizaje, semejando la realidad de forma entretenida. Ejemplo de simuladores son los Laboratorios Virtuales
- ✓ Sistemas expertos. Programa de conocimiento intensivo que resuelve problemas que normalmente requieren de la habilidad humana. Ejecuta muchas funciones secundarias de manera análoga a un experto, por ejemplo, preguntar aspectos importantes y explicar razonamientos.
- ✓ Sistemas tutoriales inteligentes de enseñanza. Pueden detectar errores, clasificarlos, y explicar por qué se producen, favoreciendo así el proceso de retroalimentación del estudiante.

- ✓ Juegos educativos. Su objetivo es llegar a situaciones excitantes y entretenidas, sin dejar en ocasiones de simular la realidad.

Por otra parte Cuba, con más de 20 años de experiencia en la construcción de este tipo de software, enfrenta actualmente el desafío de las TIC, lo que requiere de cambios cualitativos profundos en el sistema educativo cubano para continuar elevando la calidad de la educación para todos. El uso del software educativo como medio de enseñanza constituye un elemento de vital importancia para el logro de los objetivos planteados en las transformaciones de la escuela cubana y en un período relativamente corto el MINED ha organizado el proceso de producción del software educativo para todos los niveles de educación. Algunos ejemplos que evidencian el avance en esta área son los SWE (5) (6) (7): Picomaestro, A Jugar, InfoHis, entre otros.

1.1.2-Proceso de enseñanza-aprendizaje

De igual forma otro de los conceptos a abordar en la presente investigación es el proceso de enseñanza-aprendizaje. Este concepto está muy manejado y estudiado por diferentes autores tanto del plano internacional como nacional. Álvarez Zayas (1998) se refiere al proceso de enseñanza –aprendizaje, como *“el movimiento de la actividad cognoscitiva de los alumnos bajo la dirección del maestro, hacia el dominio de los conocimientos, las habilidades, los hábitos y la formación de una concepción científica del mundo”* (8). Se considera que en este proceso existe una relación dialéctica entre profesor y estudiante, los cuales se diferencian por sus funciones; el profesor debe estimular, dirigir y controlar el aprendizaje de manera tal que el alumno sea participante activo, consciente en dicho proceso, o sea, "enseñar" y la actividad del alumno es "aprender".

Al mismo tiempo, algunos autores (9) (10) (11) consideran como componentes del proceso de enseñanza-aprendizaje a: los objetivos, el contenido, los métodos, los medios y su organización, los que conforman una relación lógica interna. No obstante uno de los elementos fundamentales del proceso es la relación sistémica entre el profesor, el alumno y el contenido como el triángulo de interacción clave para la obtención del conocimiento.

1.1.3-Hiperentorno de Aprendizaje

Los hiperentornos de aprendizaje es un concepto que nace con la revolución y desarrollo de las TIC en la pasada década. Variada es la literatura (12) (13) que se encuentra del análisis realizado tanto en Cuba

como en el mundo sobre la definición de esta terminología. Según M. Sc. Orestes Coloma un hiperentorno de aprendizaje es considerado *“una mezcla de diferentes tipos de SWE, sustentados en la tecnología hipermidia, constituyendo un modelo medio de enseñanza para el apoyo al proceso de enseñanza – aprendizaje. Estos modelos constituyen un fundamento teórico y conceptual del desarrollo de las colecciones cubanas de SWE”* (12). Además, estos hiperentornos se caracterizan por:

- ✓ Ser elaborados a partir de necesidades de la enseñanza y por encargo de las mismas, o sea, a partir del conocimiento de los problemas de cada una de las asignaturas y con la participación de los mejores especialistas en la materia.
- ✓ Carácter curricular extensivo, es decir, dan respuesta a todo el currículum de una determinada asignatura en un grado o nivel.
- ✓ Compuestos por varios módulos para dar respuesta a su empleo en los diferentes momentos del proceso de enseñanza aprendizaje (introducción del nuevo contenido, ejercitación, evaluación, etc.)
- ✓ Dentro de estos módulos se incluyen aquellos pensados para el maestro o profesor y para la dirección y control del proceso de enseñanza-aprendizaje.
- ✓ Permiten el trabajo grupal, no solo para favorecer el empleo de los mismos por varios estudiantes sino para aprovechar el valor del aprendizaje colaborativo.
- ✓ Incorporación de recursos multimedia como imágenes fijas y en movimiento, sonidos y vídeos.
- ✓ Empleo de estrategias pedagógicas de análisis de respuestas y retroalimentaciones reflexivas o niveles de ayuda.
- ✓ Incorporación de servicios informáticos adicionales de búsqueda, exportación de información, impresión, música, etc.

1.1.4-Plataformas Educativas

Según Covadonga (2010), las Plataformas Educativas, *“son entornos virtuales de educación y aprendizaje (EVEA), que surgen y se desarrollan con la intención de dar soporte a la docencia, se apoyan en sistemas informáticos basados habitualmente en protocolos www, con herramientas adaptadas a las necesidades educativas requeridas. Esos entornos virtuales implican una renovación en los procedimientos de enseñanza y de aprendizaje, la manera de presentar la información de forma dinámica, que puede*

modificarse constantemente, de ahí el interés de trabajar con metodologías modulares, y la capacidad de integrar medios distintos, rompiendo con la noción de enseñanza centrada en el profesor para orientarse hacia una interdisciplinariedad basada en el alumno” (14).

Agudelo (2006), propone la siguiente definición: *“Las Plataformas Educativas permiten estimular la idea de cooperación y de interacción, como aspectos centrales del proceso de aprendizaje y enseñanza, mediante el uso de herramientas colaborativas que favorecen la adquisición de aprendizajes significativos en los estudiantes y que al mismo tiempo afianzan en las docentes prácticas de enseñanza mediadas por las Tecnologías de la Información y la Comunicación” (15).*

De manera general las plataformas educativas favorecen a la comunicación entre el estudiante y el profesor a través de un vínculo más ameno e interactivo. Los estudiantes pueden reflejar sus inquietudes, dudas y criterios acerca de los temas que se brindan en estas plataformas, ayuda a establecer el vínculo interactivo PROFESOR-ESTUDIANTE-CONTENIDO. Además constituyen un entorno virtual que acoge variadas herramientas para fines docentes. Su función es permitir la creación y gestión de cursos que permitan elevar el nivel de conocimiento de sus usuarios. En sentido general, las plataformas educativas favorecen sin lugar a dudas el proceso de aprendizaje de los estudiantes y entre otras características tiene como factor clave que los alumnos pueden intercambiar entre ellos a través de los debates, foros, chats. Constituyen un espacio más donde los estudiantes pueden dirigirse en caso de que se les presenten dudas en algún tema de investigación.

1.1.5-Laboratorios Virtuales

En la actualidad, los procesos tradicionales de enseñanza – aprendizaje se apoyan en la utilización de disímiles tecnologías educativas como la Realidad Virtual. Durante los últimos años, las aulas se han estado convirtiendo en espacios virtuales donde el ordenador es el medio por el cual se expone la información, aumentando así los niveles de percepción, atención y memoria en el procesamiento de los contenidos. El desarrollo vertiginoso de dicha tecnología ha propiciado que se adecuen diversas potencialidades que esta brinda al desarrollo de laboratorios virtuales como apoyo al proceso de enseñanza-aprendizaje.

Los laboratorios virtuales los definen algunos autores como: *“simulaciones de prácticas manipulativas que pueden ser hechas por la/el estudiante lejos de la universidad y el docente” (16); “un centro sin paredes cuyos usuarios pueden investigar sin tener en cuenta su situación geográfica interactuando con los*

colegas, teniendo acceso a los instrumentos, compartiendo los datos y los recursos informáticos, [y] recurriendo a la información de las bibliotecas electrónicas. Ese entorno se apoya en unos programas informáticos que permiten trabajar en colaboración y simultáneamente a diversas personas desde distintos sitios” (17).

Según el informe de la reunión de expertos realizado en la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura en el año 1999, los laboratorios virtuales constituyen un espacio electrónico de trabajo concebido para la colaboración y la experimentación a distancia, con el propósito de investigar o realizar otras actividades creativas, las cuales se sustentan en las nuevas tecnologías de la información y las comunicaciones. Generalmente son creados con objetivos muy parecidos, y por ende sus funcionalidades están basadas en las mismas actividades. Aunque de cualquier forma se diferencian mucho dado la cantidad de opciones o perfiles que pudieran requerir las facilidades que estos brindan, como por ejemplo: (18)

- ✓ Centrados en proyectos concretos o actividades específicas.
- ✓ La distribución geográfica de sus participantes o alumnos no tiene que ser la misma; pueden estar en distintas partes del planeta.
- ✓ Permite compartir conocimientos e informaciones acerca del marco de trabajo y/o estudio en que se encuentren.

En resumen, están diseñados para el aprendizaje de materias, temas y/o consulta de información necesaria, para elevar el nivel del ser humano moderno.

De igual manera los laboratorios virtuales están clasificados en tres tipos principalmente teniendo en cuenta el entorno en que son desarrollados:

- ✓ Laboratorios virtuales software: Son aquellos que son desarrollados como un programa de software independiente destinado a ejecutarse en la máquina del usuario, y cuyo servicio no requiere de un servidor Web. Es el caso de programas con instalación propia, que pueden estar destinados a plataformas Unix, Linux, M.S. Windows, etc. e incluso necesitar que otros componentes de software estén instalados previamente, pero que no necesitan los recursos de un servidor determinado (como bases de datos o Propuesta de Portales de Laboratorios Virtuales y Remotos). También determinados laboratorios virtuales pensados inicialmente como aplicaciones Java accesibles a

través de un servidor Web se pueden considerar de este tipo si funcionan localmente y no necesitan recursos de un servidor en concreto.

- ✓ Laboratorios virtuales Web: Estos se basan en un software que depende de los recursos de un servidor determinado. Esos recursos pueden ser bases de datos, software que requiere ejecutarse en su servidor la exigencia de determinado hardware para ejecutarse, etc. Estos no son programas que un usuario pueda descargar en su equipo para ejecutar de forma independiente.
- ✓ Laboratorios remotos: Los laboratorios remotos permiten operar como la palabra indica, remotamente con cierto equipamiento, bien sea, didáctico como maquetas específicas, o industrial, además de poder ofrecer capacidades de laboratorio virtual. En general, estos laboratorios requieren de equipos servidores específicos que les den acceso a las máquinas a operar de forma remota, y no pueden ofrecer su funcionalidad ejecutándose de forma local. Otro motivo que hace dependientes estos laboratorios de sus servidores es la habitual gestión de usuarios en el servidor. Estos se basan en instrumentos reales (tarjetas de adquisición de datos, instrumentos de medida, conexiones en interfaces diversas, comunicación de datos). Tienen como ventaja proporcionar un alto nivel de interactividad, en el que el alumno entra en contacto con equipamiento real, en vez de utilizar un equipamiento simulado. Son muy utilizados en la Educación, por ello, es de primer orden tener en cuenta el diseño con que sean implementados.

Nuestro país no ha estado ajeno a tales avances en el área de los laboratorios virtuales. El desarrollo y empleo de la tecnología en los procesos de aprendizaje en estudiantes universitarios, se ha logrado integrar a los Laboratorios Virtuales como parte del plan de estudio, ejemplo de esto lo muestra el laboratorio de química general de la Universidad Central de las Villas. Además el uso de los laboratorios virtuales se encuentra inmerso en diversas asignaturas a estudiar: medicina, carreras de Física, ingenierías (telecomunicaciones, informática, entre otras) e incluso, en el estudio de idiomas, en todas ellas, con el objetivo de ayudar a afianzar conocimientos y lograr que el aprendizaje de una materia sea diferente y ameno.

1.2- Análisis de soluciones existentes en el desarrollo de Laboratorios Virtuales.

En el mundo, a través del desarrollo vertiginoso de las TIC, existen diversas simulaciones de laboratorios virtuales tanto a nivel internacional como nacional. Estos laboratorios constituyen un centro de intercambio de conocimientos y formación de las personas, mediante la utilización de medios

audiovisuales que lo propicien ampliamente, como son: videos, multimedia, documentaciones digitales, herramientas y software desarrollados. Algunos de los laboratorios que se pueden mencionar y evidencian lo anteriormente expuesto son:

- ✓ **Laboratorio Virtual de Biología Celular y Genética(Virtual Biology Laboratory):** El software permite acceder a una serie de temas de biología (Microscopía, Estructura Celular, Membrana Celular, Genética, Ecología), para realizar diversas actividades experimentales ,el mismo fue desarrollado por el C5 (Centro de Computación y Comunicación para la Construcción del Conocimiento de la Universidad de Chile), el cual ofrece la oportunidad de apoyar la innovación y el cambio en la institución educativa a través de la integración curricular de las TIC en diferentes contextos escolares.
- ✓ **Laboratorio virtual de producción de moscas transgénicas(Virtual Transgenic Fly Lab):** Es un laboratorio dedicado principalmente a la familiarización con la ciencia y las técnicas usadas con las moscas transgénicas .Los organismos transgénicos, que contienen ADN que se inserta de forma experimental, se utilizan para estudiar muchos procesos biológicos. En este laboratorio, se creará una mosca transgénicos para estudiar los ritmos circadianos. Este laboratorio fue desarrollado en el Instituto Médico Howard Hughes (Howard Hughes Medical Institute (HHMI) de los Estados Unidos.
- ✓ El **Model ChemLab** que les proporciona a los estudiantes de Química una fuente de ese tipo de datos, accesible e intuitiva en computadora, que podría ponerse al día sistemáticamente, incluyendo nuevos datos y los cambios subsecuentes del contenido del curso. Se originó de trabajos académicos sobre simulación computacional y diseño de software en McMaster University situada en Toronto, Canadá. Este laboratorio virtual incorpora una simulación interactiva con un ambiente de trabajo con áreas independientes, tales como teoría, procedimientos y observaciones de estudiantes, además se utilizan equipos y procedimientos comunes para simular los pasos que se deben dar para realizar cierto experimento.
- ✓ **Laboratorio Virtual Química**, en la Universidad de Villa Clara “Martha Abreu”, el cual a medida que el estudiante va trabajando y avanzando le muestra los conocimientos que ha ido alcanzando en la materia. Con el desarrollo del software multimedia se logra mostrar la simulación de la práctica de laboratorio *“Influencia de la concentración y la temperatura en la velocidad de reacción del Tiosulfato de Sodio y el Ácido Sulfúrico”*, combinando elementos multimedia con Programación Orientada a

Objetos (POO). Este software se utiliza en otros Centros de Educación Superior del país, y por los estudiantes de las Sedes Universitarias Municipales. Además, presenta una ayuda de contexto que facilita su uso y una interfaz amena para los usuarios y de fácil entendimiento.

- ✓ El software **La naturaleza y el Hombre** forma parte de la **Colección El Navegante** perteneciente a las colecciones de software del MINED utilizado con el objetivo de constituir un medio de apoyo al proceso de enseñanza-aprendizaje para el nivel secundario. Este producto aborda los contenidos de las asignaturas de Biología, Física, Química y Geografía que se imparten en los tres grados de la Educación Secundaria Básica.
- ✓ En la Universidad de las Ciencias Informáticas (UCI) surgida al calor de la batalla de ideas en el 2002, por sus características de nuevo tipo de universidad, capaz de revolucionar al Sistema Educativo y forjar a los jóvenes universitarios cubanos, se trabaja en la asignatura de Física I con experimentos virtuales sustentados sobre Aplicaciones de Laboratorios Virtuales, tal es en el caso desde la etapa comprendida entre 2004-2005. Este entorno virtual de Física consta con 8 prácticas de LV para el estudio de algunos temas de Física I que favorece ampliamente a las estudiantes para alcanzar los conocimientos en la asignatura. También el “Nodo Virtual de Procesos Versión 2.0” desarrollado en la Facultad 3 en el año 2010 con el objetivo de desarrollar la simulación de procesos industriales, mediante el uso de modelos matemáticos. Esta aplicación permite a los estudiantes de la especialidad Ingeniería Automática probar estrategias de control y reforzar el proceso docente de los mismos. (19)

A partir de la investigación realizada en múltiples ámbitos es válido decir que es muy interesante el alto grado de interactividad de las interfaces de comunicación y los procedimientos que se ejecutan en los laboratorios estudiados. Además, los procesos se realizan de manera independiente, a medida que se realiza cada práctica de laboratorio, lo cual propicia una fácil navegación al usuario, al interactuar dentro de las áreas que estos contienen en su interior.

Como guía para realizar la presente investigación se utilizó el laboratorio virtual Model ChemLab, para la descripción de las funcionalidades que debe realizar el laboratorio de Biología. Las simulaciones de los procesos de este laboratorio en su primera versión estaban realizadas en dos dimensiones (2D) (20) y actualmente en su última versión está sustentado en representaciones gráficas en tres dimensiones (3D) (21), lo cual propicia un mayor acercamiento a procesos reales. También, existe un peso de trabajo en las

interfaces de comunicación desde el punto de vista funcional y desde el punto de vista de diseño gráfico que este presenta. Además de apropiarse de la tecnología multimedia e hipermedia.

Existe además una particularidad novedosa en la actual investigación. El laboratorio virtual de Biología al cual se le propone el desarrollo de un análisis y diseño será parte en un futuro de la plataforma de gestión del aprendizaje ZERA, específicamente de uno de los hiperentornos de aprendizaje que esta posee, el destinado a la materia de Biología. Además, es válido destacar que los laboratorios mencionados anteriormente representan aplicaciones web y de escritorio, pero no poseen esta característica que lo hace ser diferente a los demás.

1.3- Metodologías de desarrollo

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales en el transcurso del desarrollo del software.

Constituye además, un proceso para la producción organizada del software, empleando una colección de técnicas predefinidas y convenciones en las notaciones. Se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso. Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo.

La construcción de un software implica la toma de decisiones sobre la arquitectura del sistema (definir los componentes del sistema de software y sus interacciones). Estas decisiones pueden ser cruciales para el éxito o fracaso del sistema resultante, por lo que se requiere seleccionar un proceso de desarrollo de software, con el fin de obtener la calidad del sistema de software deseada y que cumpla con los requerimientos establecidos. Metodologías vigentes de ingeniería de software atienden muy bien estos requerimientos y permiten al equipo encargado de dicha labor asumir con propiedad su función (22).

Por la importancia que reviste para la presente investigación, se abordan algunas de las metodologías para el desarrollo de software. Las metodologías RUP y XP por ser las principales representantes de los dos enfoques en el desarrollo de metodologías. RUP la más utilizada en representación de las metodologías pesadas o tradicionales y XP por ser la metodología más manejada en el área de las

metodologías ágiles. Además, se investigó acerca de un proceso para el desarrollo de SWE en proyectos productivos de la UCI.

1.3.1-Metodología eXtreme Programing (XP)

La metodología de Programación Extrema (XP, eXtreme Programing) es uno de los ejemplos más exitosos de metodología ágil. Se centra en el trabajo orientado directamente al objetivo, basándose para esto en las relaciones interpersonales y en la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso. Una característica saltante de XP, es que el código siempre se produce en parejas, estas van cambiando constantemente para lograr así que todo el equipo sepa y pueda modificar según necesidades el código generado, esto logra en el equipo que los integrantes aprendan entre sí y compartan todo el código.

Uno de los inconvenientes que presenta es que solo se recomienda aplicar en proyectos a corto plazo. Además, en esta metodología el desarrollo va acompañado con la opinión del cliente que es un integrante más del equipo de desarrollo. XP prácticamente no genera documentación fuera del código fuente, crea muy pocos artefactos para el modelado de los procesos que se llevan a cabo, lo que dificulta la representación de los sistemas del negocio o la utilización de experiencia anteriores. Está pensado para equipos de desarrollo relativamente pequeños y para proyectos con requisitos muy cambiantes.

1.3.2-Proceso Unificado de Desarrollo de Software (RUP)

“El Proceso Unificado RUP (Rational Unified Process), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Proporciona una aproximación disciplinada a la asignación de tareas y responsabilidades. RUP actúa como modelo y puede ser adaptado y extendido”. (23)

RUP presenta tres características fundamentales:

1. **Está dirigida por los Casos de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, estos representan los requisitos funcionales. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
2. **Está centrada en la arquitectura:** La arquitectura muestra una visión común del sistema completo en la que el equipo de trabajo y los usuarios deben de estar de acuerdo, por lo que describe los

cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

3. **Es iterativa e incremental:** Propone que cada fase se desarrolle en iteraciones. Una iteración implica actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

La metodología RUP, proporciona disciplinas en las cuales se generan artefactos que permiten documentar e implementar de una manera eficiente el software a desarrollar. Además constituyen guías para un buen desarrollo, dentro de las fases que cuenta esta metodología podemos mencionar las siguientes:

- ✓ **Concepción o inicio:** Comprender los requisitos y determinar visión y alcance del proyecto.
- ✓ **Elaboración:** Asignar recursos, especificar las características y definir la arquitectura.
- ✓ **Construcción:** Implementación, construir el producto operacional.
- ✓ **Transición:** Hacerlo operativo para los usuarios, nivel correcto de calidad para entregar.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales.

Los seis primeros son los flujos de ingeniería, estos son:

- ✓ **Modelamiento de negocio:** Se entienden las necesidades del negocio.
- ✓ **Requerimientos:** Se trasladan las necesidades del negocio a un sistema automatizado.
- ✓ **Análisis y diseño:** Se trasladan los requerimientos dentro de la arquitectura de software.
- ✓ **Implementación:** Se crea un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ✓ **Prueba (Testeo):** Se debe asegurar que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- ✓ **Instalación del software.**

Los tres últimos son de apoyo:

- ✓ **Administración del proyecto:** Para el control de los horarios y recursos.
- ✓ **Administración de configuración y cambios:** Permite guardar todas las versiones del proyecto.
- ✓ **Ambiente:** Se realiza la administración del ambiente de desarrollo.

RUP se basa mucho en la documentación, que aunque se ve afectada con los posibles cambios volátiles que los clientes soliciten en cuanto a funcionalidades del software, representa una gran ventaja, ya que gracias a su plan de desarrollo se pueden reconocer los problemas y fallos de forma temprana y corregirlos. Pero además, RUP no es un sistema con pasos firmemente establecidos, sino una metodología adaptable al contexto y necesidades, característica que lo hace ampliamente utilizable en áreas de desarrollo de software.

1.3.3-Proceso para el desarrollo de proyectos de SWE en la UCI.

En el 2009 se desarrolló el proceso de desarrollo para SWE en la UCI (24) respondiendo a las necesidades que existían en los polos productivos de esta esfera, una de las razones que originó su desarrollo fue la falta de una guía que identifique actividades, roles y artefactos que se ajusten a las características y necesidades de los proyectos de SWE que afectaba la capacidad de organizar y controlar adecuadamente el proceso de desarrollo, lo que podría afectar la calidad de este y hacer menos eficiente la obtención de un producto final con la calidad requerida.

El proceso, posee una estructura como la mostrada en el anexo 30, donde este se inicia con la actividad de Modelado Conceptual y luego el Diseño Instruccional enfocado a tener como resultado un SWE. Una vez concluido este proceso se inicia la Gestión de requisitos donde evolutivamente se contemplan los requisitos del sistema, del diseño pedagógico y diseño gráfico, dando paso a los procesos Análisis y Diseño del Sistema, Diseño Gráfico y Gestión Audiovisual, que pueden comenzar a ejecutarse en paralelo. En el proceso de diseño se define la arquitectura del sistema y está relacionado con el proceso de Diseño Gráfico por la posibilidad de este de modificar requisitos. Con el proceso de Gestión Audiovisual se obtienen los recursos audiovisuales que se incorporarán en el software. Dependiente del Diseño de Sistema se inicia el proceso de Implementación donde se desarrolla el sistema. Finalmente, se realizan los procesos de Prueba, que valida funcionalidad e impacto en el aprendizaje y el de Implantación, en los casos que lo requieran.

Uno de los inconvenientes que posee este proceso es que a pesar de estar validado como una propuesta de metodología, aún no está evaluado en el desarrollo real de un proyecto de SWE. De igual forma sería muy importante validar su incidencia en la organización y el control de los proyectos, así como realizarle las mejoras necesarias en este sentido.

1.4- Arquitectura y Patrones

“En la última década, cambió la visión que los desarrolladores tienen de los sistemas de software. Se llamó <<arquitectura>>. La arquitectura de software, básicamente indica la estructura, funcionamiento e interacción entre las partes del software” (25). Desde los pequeños programas hasta los sistemas más grandes, poseen una estructura y un comportamiento que los hacen clasificables según su arquitectura.

El diseño arquitectónico define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos. Los patrones expresan el esquema fundamental de organización para sistemas de software. *“Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos; así como ayudan a especificar la estructura fundamental de una aplicación”* (26).

1.4.1-Patrón arquitectónico Modelo Vista Controlador

Modelo Vista Controlador (MVC en adelante) surgido inicialmente por la comunidad de SmallTalk-80 es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- ✓ **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- ✓ **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- ✓ **Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón y pulsaciones de teclas. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente: (26)

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón: enlace)
2. El controlador recibe (por parte de los objetos de la interfaz - vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aún así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Desde el surgimiento de MVC y con el paso de los años, se han desarrollado diferentes variantes que se ajustan a las características del software educativo, entre la que se encuentra: (1) la modificada para Aplicaciones Multimedia, conocida como MVCmm, la cual diversifica las funcionalidades del Modelo, teniendo en cuenta las características de las aplicaciones Multimedia, donde tienen un gran peso las clases que representan los medios de comunicación (texto, imagen, sonido, animación) empleados y (2) MVC-E una propuesta de diseño arquitectónico contextualizada a las aplicaciones educativas cubanas (26).

1.4.2-Patrón arquitectónico Modelo Vista Controlador Entidad (MVC-E)

Las aplicaciones educativas cubanas, explotan grandemente los conceptos del Entorno Hipermedia, así como incorporan de forma profunda, las técnicas y conocimientos de bases de datos relacionales y la

Programación Orientada a Objetos más avanzada que se utiliza hoy día, como consecuencia de la necesidad de implementar diversos y complejos métodos pedagógicos desarrollados por educadores en las distintas ramas de la enseñanza cubana. Por esta condición, los diseñadores buscan y aplican soluciones ya trabajadas internacionalmente y con gran número de aplicaciones prácticas en este sentido, dentro de las que se encuentra el empleo del Patrón MVC. No obstante, este patrón tal y como está descrito actualmente, aún con su variante más actual, no responde completamente a las características mencionadas.

El patrón Modelo Vista Controlador Entidad (MVC-E) es una variante de solución al MVCmm para las aplicaciones educativas cubanas, donde se descarguen las responsabilidades de la clase modelo concernientes al procesamiento y almacenamiento de la información persistente de las aplicaciones, incorporando una nueva clase al modelo denominada *Modelo-Entidad*, con dos tipos fundamentales, la clase *Modelo-Entidad-Media* y *Modelo-Entidad-Persistente*. La primera de estas con la responsabilidad de agrupar las clases que identifican las medias y su árbol de jerarquía en la aplicación y la segunda tiene como responsabilidad la gestión de la información persistente, que antes sobrecargaba a la clase *Modelo* del patrón MVC original. Esta variación a su vez sustenta las características actuales de los sistemas multimedia como son: comunicación con bases de datos, archivos XML, o sistemas externos.

1.4.3-Patrones de diseño

Los Patrones de Diseño son directrices y principios estructurados que describen un problema común y entregan una buena solución ya probada a la que le dan un nombre.

Dentro de las principales ventajas que trae el uso de estos patrones están:

- ✓ Ayudan a diseñar correctamente en menos tiempo.
- ✓ Ayudan a construir problemas reutilizables.
- ✓ Facilitan la documentación.

Existe un grupo de patrones de diseño denominado Patrones de Principio Generales para la Asignación de Responsabilidades (en sus siglas en inglés GRASP) los cuales constituyen los principios fundamentales con que se asignan las responsabilidades a los objetos. Dentro de este grupo se encuentran los patrones de tipo: Experto, Creador, Alta Cohesión, Bajo Acoplamiento, Controlador,

además de otros como Polimorfismo, Fabricación Pura, Indirección y No Hables con Extraños. En la tabla 1.1 se presenta una muestra de estos.

Tabla 1.1 Patrones generales de software para asignar responsabilidades tomado de (27).

Patrón	Descripción
Experto	¿Quién asumirá la responsabilidad en el caso general? Asignar una responsabilidad al experto en información: la clase que posee la información necesaria para cumplir con la responsabilidad.
Creador	¿Quién crea? Asignar a la clase B la responsabilidad de crear una instancia de clase A, si se cumple una de las siguientes condiciones: 1. B contiene A 4. B registra A 2. B agrega A 5. B utiliza A muy de cerca 3. B tiene los datos de inicialización de A
Controlador	¿Quién administra un evento del sistema? Asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente una de las siguientes opciones: 1. El negocio o la organización global (un controlador de fachada). 2. El "sistema" global (un controlador de fachada). 3. Un ser animado del dominio que realice el trabajo (un controlador de papeles). 4. Una clase artificial (Fabricación Pura) que represente el caso de uso (un controlador de casos de uso).
Bajo Acoplamiento	¿Cómo dar soporte a poca dependencia y a una mayor reutilización? Asignar las responsabilidades de modo que se mantenga bajo acoplamiento.
Alta Cohesión	¿Cómo mantener controlable la complejidad? ¿Asignar las responsabilidades de modo que se mantenga una alta cohesión?

1.5- Lenguajes de modelado

En las disciplinas ingenieriles, la construcción de modelos es imprescindible, ya que ayuda al usuario a entender el sistema y la posibilidad de realizar análisis antes de invertir y gastar grandes cantidades de

recursos en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que constituirán el modelo. Estos ofrecen planos detallados que brindan una visión global del sistema en consideración.

En el caso de la Ingeniería del Software no se está ajeno a tales beneficios que aportan los modelos. Al tratarse de un sistema software (sea de cualquier naturaleza como las aplicaciones educativas) toman una connotación diferente: *“deben ser capaces de representar la información que el software transforma, la arquitectura y las funciones que permite que ocurra la transformación, las características que desean los usuarios y el comportamiento del sistema conforme se realiza la transformación”* (28).

Precisamente para especificar estos modelos de software se necesita de un lenguaje de modelado: una tecnología, que a partir de un conjunto estandarizado de símbolos, reglas y notaciones permite la representación de las características estructurales y de comportamiento de los sistemas informáticos. Seguidamente se muestran algunas propuestas importantes, de las que se hará una breve descripción y se mostrarán sus principales características.

1.5.1-Lenguaje Unificado de Modelado (UML).

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Surgido a finales de la década de los 90, constituye un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, convirtiéndose actualmente en una de las mejores herramientas para el diseño y desarrollo de software fiable, eficiente y de calidad.

Cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas:

Diagramas de estructura estática:

- ✓ Diagrama de clases: Clases, interfaces y colaboraciones; así como sus colaboraciones.
- ✓ Diagrama de objetos: Objetos y sus relaciones.
- ✓ Diagrama de casos de uso: Casos de uso, actores y sus relaciones.

Diagramas de comportamiento:

- ✓ Diagramas de interacción (secuencia y colaboración): Objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.

- ✓ Diagrama de estados: Muestra una máquina de estado que consta de estados, transiciones, eventos y actividades.
- ✓ Diagrama de actividad: Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Diagramas de implementación:

- ✓ Diagrama de componentes: Organización y las dependencias entre un conjunto de componentes.
- ✓ Diagrama de despliegue: Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

Los objetivos de UML son muchos, pero se pueden sintetizar las siguientes funciones: (29)

- ✓ **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

A pesar de que UML permite modelar casi cualquier sistema orientado a objetos, no soporta todos los aspectos de las aplicaciones multimedia de una forma adecuada e intuitiva. Especialmente, las características del lenguaje para modelar los aspectos de la interfaz de usuario, no se aplican explícitamente en los entornos multimedia. Fue entonces que gracias a las facilidades de extensión que brinda UML, surge el Lenguaje Orientado a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA-L), que facilita el modelado de un gran rango de aspectos de aplicaciones multimedia interactivas de una forma integrada y comprensiva. (30)

1.5.2-Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L)

En medio de una búsqueda para una modelación adecuada de las aplicaciones multimedia surge en el 2001 por Stefan Sauer y Gregor Engels, en la Universidad de Paderborn, Alemania, el Lenguaje de

Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L) (31) como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y en el patrón arquitectónico MVC (original) para la interfaz de usuario. (32)

Dicho lenguaje está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son:

- ✓ **Vista Lógica:** Modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.
- ✓ **Vista de Presentación espacial:** Modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, y separa los objetos de visualización e interacción.
- ✓ **Vista de Comportamiento temporal predefinido:** Modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo.
- ✓ **Vista de Control Interactivo:** Modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sintácticamente igual a este último mas con la diferencia semántica de que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

El lenguaje OMMMA-L soporta el modelado de los aspectos estructurales, funcionales y dinámicos de un sistema interactivo y su interfaz de usuario. Además de concentrarse en la funcionalidad desde la perspectiva del sistema de software, su sintaxis es definida explícitamente puesto que su semántica es informal e intuitiva. Es por ello que la extensión del lenguaje notacional UML a OMMMA-L para el tratamiento de aplicaciones educativas multimedia en los últimos años ha sido de gran aceptación.

Este lenguaje presenta como inconveniente que no hace una especificación formal de los elementos que extiende de UML, por ejemplo la definición de estereotipos que restrinjan el concepto clase del UML para la representación de los medios de comunicación que se describen.

1.5.3-El Lenguaje de Modelado orientado a objeto para Aplicaciones Educativas y Multimedia (ApEM-L)

El Lenguaje de Modelado orientado a objeto para las Aplicaciones Educativas y Multimedia (ApEM-L) surge su primera versión en el 2007 con motivo de dar una nueva solución al modelado de aplicaciones educativas multimedia en el entorno productivo de la Universidad de las Ciencias Informáticas. El mismo se presenta como una extensión de UML y tomando como bases teóricas principales OMMMA-L (2001) y respetando el estándar OCL¹-2.0 (2003), lo que hace que el lenguaje sostenga sólidas bases científicas y de uso de estándares mundiales.

La concepción del ApEM-L tuvo los siguientes objetivos: (30)

- 1) Desarrollar una extensión del lenguaje de modelado UML, tomándolo como base e incorporando a este, a través de sus mecanismos de extensión, los elementos fundamentales del proceso productivo UCI. De esta forma se produjo un lenguaje de propósito particular para la modelación de aplicaciones educativas.
- 2) Incorporar los elementos más significativos de extensiones anteriores como OMMMA-L y a su vez respetar lo establecido por el estándar OCL, para de esta forma lograr una extensión consistente y escalable en el tiempo.
- 3) No complicar ApEM-L con elementos que lo convirtieran o abarcaran en un método de desarrollo de aplicaciones educativas, sino solo el área de la representación y la documentación de este tipo de aplicaciones.
- 4) No circunscribir ApEM-L a un proceso de desarrollo en específico, sino expresarlo de manera tal que pueda ser utilizado con cualquiera de los existentes, aunque se sugiere la utilización de procesos de

¹ El Lenguaje restringido a objetos (OCL en sus siglas en inglés) es un lenguaje de especificación formal para lograr una mayor precisión a las especificaciones de modelado orientada a objeto. Dispone de todo el poder de la lógica y la matemática discreta. (44)

desarrollos iterativos, incrementales y basados en prototipos, que permitan la modelación de sistemas orientados a objetos.

Los conceptos y modelos de ApEM-L en su primera versión se agrupan en las siguientes áreas conceptuales:

- ✓ **Estructura lógica.** Esta área se compone de la vista estática y la vista de arquitectura. La primera de ellas se compone por el diagrama de clases y el diagrama de casos de uso los cuales no sufren ninguna extensión. A su vez la vista de arquitectura la componen el diagrama de componentes y el diagrama de despliegue. De los diagramas mencionados solo han sido modificados los siguientes: diagrama de clases y el diagrama de componentes, el resto mantuvo lo establecido por UML. Para los diagramas de casos de uso, se adicionaron un conjunto de elementos a la descripción textual de los casos de uso propuesta por UML, para una mejor descripción del contexto productivo de los software educativos. Además se tomaron como base las ideas expresadas por (31), mejoradas al contexto productivo UCI con la incorporación del modelo arquitectónico MVC-E.
- ✓ **Comportamiento dinámico.** Realmente esta área no sufre grandes modificaciones, pues solo se hace una pequeña adición al diagrama de secuencia, salvo que ciertamente se enriquece de la semántica original de UML para estos diagramas. Tal modificación se basa en la adición de una variable de tiempo en cualquier lugar que sea necesario su especificación para un mejor entendimiento
- ✓ **Gestión del modelo.** Esta área es la que sufre grandes cambios tanto en su carácter semántico como sintáctico, con la incorporación de estereotipos restrictivos en todos los diagramas a partir de nuevos conceptos incorporados a los diagramas de clases originales o básicas de UML. En la misma se crean dos nuevos diagramas: el de *estructura de la presentación* y el de *estructura de la navegación*.

A raíz de la puesta en marcha en el 2008 del lenguaje de modelado ApEM-L 1.0 en entornos productivos de instituciones que hoy día son productoras del SWE, se prueba de alguna manera el marco tecnológico base del lenguaje. Como consecuencia de esta acción surgieron en aquel momento nuevas problemáticas y necesidades en la comprensión de los artefactos por los diferentes equipos de desarrollo de dichas empresas, dando paso al surgimiento de una nueva versión del lenguaje (ApEM-L 1.5), la cual permitió

una mejora potencial para el ámbito productivo e ingenieril del software educativo cubano. Tales modificaciones son: (1)

- ✓ Definición del principio de organización del sistema de ApEM-L y el surgimiento del artefacto *Entorno de Presentación* el cual muestra la descripción estructural de los elementos de la interfaz de comunicación; así como su relación, para el cumplimiento de una determinada función del sistema. Este artefacto es un componente abstracto de agrupamiento, que ha sido incorporado al lenguaje base como extensión del UML, enriqueciendo la sintaxis y la semántica del concepto *clase*. El mismo, responde a los requerimientos funcionales de la aplicación y representa a las clases Vistas del patrón MVC, que tienen como responsabilidad, el tratamiento de las entradas y las salidas de información del software. En el SWE, realizan el tratamiento de los antiguos escenarios o pantallas. Según esta definición y por constituir la base general del modelo se define desde el punto de vista estructural que una interfaz de usuario se compone por varios Entornos de Presentación y este último a su vez puede agregar otros en si tal como se muestra en la figura del anexo 15. Además toda interfaz gráfica de usuario es al mismo tiempo un entorno de presentación por el mero hecho de que puede contener todos los otros entornos de presentación en su interior.
- ✓ Prescindir del uso de la vista de Caso de Uso debido a que el Entorno de Presentación desde una filosofía ingenieril ocupaba el espacio del mismo, al expresar igual los requerimientos funcionales y no funcionales del sistema.
- ✓ Mejora de la especificación textual de Casos de Uso presentado por ApEM-L 1.0 precisamente para dejar claro y detallado todo el flujo de eventos; así como todas las características esenciales y necesarias que engloba los entornos de presentación y el SWE.
- ✓ Incorporación del artefacto Actor al Diagrama de Estructura de Navegación por ser un elemento fundamental en el inicio de la navegación del software.
- ✓ Incorporación del estereotipo para el tratamiento de la condición de transición entre las clases vistas en el DEN.

1.6- Herramientas de modelado

Las herramientas de modelado de sistemas informáticos, son herramientas que se emplean para la creación de modelos de sistemas que ya existen o que se desarrollarán. Las herramientas de modelado,

permiten crear un "simulacro" del sistema, a bajo costo y riesgo mínimo. A bajo costo porque, al fin y al cabo, es un conjunto de gráficos y textos que representan el sistema, pero no son el sistema físico real (el cual es más costoso). Además minimizan los riesgos, porque los cambios que se deban realizar (por errores o cambios en los requerimientos), se pueden realizar más fácil y rápidamente sobre el modelo que sobre el sistema ya implementado.

Las buenas herramientas de modelado cumplen con determinadas características:

- ✓ Permiten una visión descendente del sistema.
- ✓ Permiten particionar el sistema.
- ✓ Poseen componentes gráficos con algo de apoyo textual.
- ✓ El modelo resultado debe ser fácil de comprender.
- ✓ Poseen mínima redundancia.

A continuación, se analizarán dos de las herramientas de modelado más usadas a nivel mundial, como resumen del estudio del arte realizado.

1.6.1-Argo UML

ArgoUML es una herramienta para modelar sistemas, mediante el cual se realizan diseños en UML "Lenguaje de Modelado Unificado", basada en Java. Puede crear la mayoría de los diagramas estándares de UML. Además fue concebido como una herramienta para el análisis y el diseño de los sistemas de software orientados al objeto. En este sentido es similar a muchas de las herramientas comerciales CASE (Computer Aided Software Engineering, en inglés) que se venden como herramientas para modelar sistemas de software.

ArgoUML es un proyecto abierto. La disponibilidad del código fuente asegura de que una nueva generación de diseñadores y de investigadores de software tenga un marco probado del cual puedan conducir el desarrollo y la evolución de las tecnologías de la herramienta CASE.

1.6.2-Visual Paradigm

Visual Paradigm es una herramienta UML profesional fácil de usar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El

software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Además, permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Dentro de las principales características de la herramienta están:

- ✓ Soporte de UML versión 2.1.
- ✓ Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio.
- ✓ Diagramas de flujo de datos.
- ✓ Generador de informes para generación de documentación.
- ✓ Distribución automática de diagramas.
- ✓ Importación y exportación de ficheros.
- ✓ Editor de figuras.

Visual Paradigm ofrece un entorno amigable para el usuario, sugiere nuevos posibles componentes a utilizar, por lo que ya no es necesario localizarlos en la barra y así se crea fácilmente cualquier tipo de diagrama. Incluye gran variedad de estereotipos para la creación de diagramas de fácil entendimiento, los que organiza automáticamente.

1.7- Lenguaje de Desarrollo

1.7.1 -ActionScript 3.0

ActionScript es el lenguaje de programación de la Plataforma Adobe Flash. Originalmente desarrollado como una manera para los desarrolladores programar interactivamente. ActionScript permite la programación eficiente de aplicaciones de la Plataforma Flash, desde animaciones simples hasta complejas, ricas en datos e interfaces interactivas (33)

Se utilizará para futura implementación del componente, el comportamiento de los objetos en el entorno interactivo y las clases necesarias para su implementación.

1.8- Framework de Desarrollo

1.8.1 - Flex SDK 4.0

Flex SDK provee un paquete de herramientas para el desarrollo de aplicaciones enriquecidas de internet tales como: lenguaje de marcas MXML (Macromedia Extensible Markup Language) basado en XML, un conjunto de librerías y componentes visuales escritos en lenguaje ActionScript 3.0 para facilitar el trabajo a los desarrolladores así como un compilador para este. (34)

Al permitir la creación de aplicaciones en lenguaje ActionScript 3.0 proporciona una gran facilidad de integración con otros framework de desarrollo como es el caso de motores gráficos 3D escritos en dicho lenguaje, con el objetivo de desarrollar escenarios virtuales tridimensionales tanto para la web como en aplicaciones de escritorio. (34)

1.9- Propuesta de solución técnica

El proceso de selección de las tecnologías y herramientas para el desarrollo de la aplicación, implicó un estudio por el carácter vital y la influencia posterior que suponen las decisiones a adoptar para el desarrollo del análisis y diseño del laboratorio virtual de Biología. Teniendo en cuenta las metodologías analizadas finalmente se escoge como metodología de desarrollo a RUP, debido a que es una metodología abierta y adaptable al desarrollo de software educativo, lo cual garantiza que se lleven a cabo sólo aquellas actividades y modelos que sean necesarios o útiles para el proyecto a desarrollar. El hecho de que RUP sea iterativo e incremental, garantiza que en cada iteración se va obteniendo un producto más acabado. La ventaja principal de esta metodología es que se basa en las mejores prácticas que se han intentado y se han probado en el campo del desarrollo de software. Además, RUP basa su trabajo principalmente en la documentación del software y expone un conjunto de actividades orientadas a visualizar, especificar, construir y documentar los artefactos necesarios para el desarrollo de software con la calidad que el cliente necesita y es mayormente utilizado en equipos grandes de trabajo. Como lenguaje de modelado para la solución propuesta se hará uso del lenguaje de modelado ApEM-L ya que después de haber analizado algunos lenguajes de modelado de software como UML y su extensión OMMMA-L, se ha llegado a la conclusión, que a pesar de que en la última década se ha logrado incorporar estos lenguajes notacionales en el modelado de entornos educativos, no logran denotar todos los elementos, tanto ingenieriles como pedagógicos de las aplicaciones educativas cubanas en sus gráficos y semánticas utilizadas.

Por su parte ApEM-L surge con motivo de dar una nueva solución al modelado de aplicaciones educativas multimedia y dentro de sus principales ventajas están:

- ✓ Puede utilizar para su representación todas las herramientas CASE que existen actualmente para el modelado de UML.
- ✓ Es un lenguaje que utiliza el estándar internacional OCL, para la modelación de la programación Orientada a Objetos.

En la presente investigación realizada se analizó además del Visual Paradigm, la herramienta CASE Argo UML pero se escogió el Visual Paradigm como herramienta modelado debido a que tiene disponibilidad en múltiples plataformas, ofrece apoyo adicional en cuanto a generación de artefactos automáticamente, permite dibujar todos los tipos de diagramas usados en el proceso de desarrollo de software que utiliza RUP y permite la utilización de la ingeniería inversa, utilizando el código para obtener los modelos y los diagramas. Además, debido a que es una herramienta libre, permite la modelación de software usando las versiones más actualizadas de UML, lenguaje que constituye la base arquitectónica de ApEM-L.

Conclusiones parciales

En este capítulo se realizó un estudio de los principales conceptos relacionados con el dominio del problema. Además, se abordó acerca de características actuales de los laboratorios virtuales tanto en el ámbito nacional e internacional, como en las universidades de Cuba y se realizó el estudio de las tecnologías a utilizar en el desarrollo de la propuesta de solución, donde se escogió la metodología RUP. Como lenguaje de modelado se utilizará ApEM-L y la herramienta para la representación del software que sirve como apoyo automático a los diferentes modelos que se obtendrán en el desarrollo de la solución y que ayudará con el desarrollo del análisis será Visual Paradigm. Además es importante mencionar que se hizo referencia a la arquitectura de software utilizada en todas sus variantes, destacando el patrón arquitectónico Modelo-Vista-Controlador-Entidad (MVC-E) variante de modificación al Patrón MVC como propuesta de diseño arquitectónico a las aplicaciones educativas cubanas. A lo largo de este capítulo se han ofrecido los elementos teóricos que sirven de sustento científico a la investigación, así como cada uno de los aspectos a tener en cuenta en la situación problemática que genera el problema científico razón de esta investigación.



CAPÍTULO II: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Introducción

En el presente capítulo se realiza una descripción de la solución que se propone para el problema identificado. Se modela el entorno donde se usará el software y se identifican y describen sus principales conceptos asociados, reflejados mediante la realización de un Modelo del Dominio. Se definen los requerimientos funcionales y no funcionales por los que se regirá el software propuesto. Se representa además, la Vista de Gestión del Modelo y sus subsistemas. Se hace una descripción textual de los Entornos de Presentación identificados. Además se muestran los Diagramas de Estructura de Presentación y los Diagramas de Estructura de Navegación como artefactos propuestos por el lenguaje de modelado utilizado: ApEM-L.

2.1 - Modelado del Dominio

Por muy pequeño que sea un sistema, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad (principio ingenieril de modularidad). Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Debido a que para la modelación del laboratorio de Biología propuesto no se lograron definir procesos específicos en el entorno del negocio, se decide identificar conceptos y objetos relacionados en dicho entorno y representar estos a través de un Modelo del Dominio (conocido también como Modelo Conceptual). El Modelo del Dominio *“es una representación visual que captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan <<las cosas>> que existen o los eventos que suceden en el entorno en el que trabaja el sistema”* (23). Además es una de las alternativas que brinda RUP para la identificación de requisitos y la comprensión del contexto cuando existe poca estructuración en los procesos de negocio.

2.1.1 -Análisis de los conceptos del dominio.

Para lograr una mayor comprensión del diagrama de Modelo de Dominio que se expondrá a continuación, es necesario realizar previamente la definición de los conceptos que involucra dicho modelo.

- ✓ **Estudiante:** Persona que cursa estudios, tiene el propósito realizar una práctica de laboratorio, accede al hiperentorno en busca de conocimientos.
- ✓ **Docente (Profesor):** Persona que asigna las prácticas de laboratorio a los estudiantes, encargado de hacer seguimiento a las actividades de todos los estudiantes que estén a su cargo.
- ✓ **Laboratorio:** Espacio donde se realizan las prácticas de laboratorio.
- ✓ **Práctica de Laboratorio:** Tipo de clase que tiene como objetivo instructivo adquirir habilidades a través de la experimentación.
- ✓ **Ayuda:** Guía que contiene información de los materiales e instrumentos a utilizar en una práctica de laboratorio.
- ✓ **Diagnóstico Previo:** Evaluación realizada al estudiante previamente antes de acceder al laboratorio acerca del contenido de la práctica a realizar.
- ✓ **Ejercicio:** Actividad mental que consiste en la repetición de ciertas rutinas, con el fin de desarrollar una determinada habilidad y adquirir conocimientos referente a un tema específico.
- ✓ **Local:** Lugar del laboratorio donde se realizan procesos y se tienen otros recursos del laboratorio.
- ✓ **L_Reactivos-MaterialesBiológicos:** Local que contiene los reactivos y materiales biológicos que se utilizan en el laboratorio para realizar la práctica.
- ✓ **L_Esterilización:** Local donde se esterilizan los recursos utilizados en las prácticas de laboratorio.
- ✓ **Norma de Uso:** Reglas que se deben cumplir en aras de mantener la protección del ambiente del laboratorio.
- ✓ **ObjetosLaboratorio:** Elementos que se encuentran dentro del laboratorio y forman parte de su entorno.
- ✓ **Pizarra:** Medio donde se expone el título, el objetivo y los recursos de la práctica de laboratorio a realizar.
- ✓ **Mesa de trabajo:** Objeto del laboratorio donde se colocan los utensilios y sobre el cual se realizan los principales procedimientos de la práctica de laboratorio. Está compuesto por cuatro estantes y un fregadero.

- ✓ **Recurso de Aprendizaje:** Medio de apoyo que aporta información relacionada a la práctica de laboratorio dígame de libros, un objeto representando una célula, el torso de un humano etc.
- ✓ **Estante:** Mueble donde se almacenan diferentes tipos de recursos de trabajo.
- ✓ **Fregadero:** Objeto del laboratorio que está situado en la mesa de trabajo en el cual se realiza el lavado de los recursos del laboratorio utilizados durante la realización de la práctica.
- ✓ **Procedimientos Técnicos:** Operaciones específicas de la práctica a realizar.
- ✓ **Paso:** Una operación específica del procedimiento técnico.
- ✓ **Recursos de la Práctica:** Utensilios que se usan como medio de trabajo para realizar los procedimientos de la práctica.
- ✓ **Equipo Fijo:** Recurso del laboratorio que se sitúa encima de la mesa de trabajo.
- ✓ **Instrumento de medición:** Recurso del laboratorio que se usa para comparar magnitudes físicas mediante un proceso de medición.
- ✓ **Miscelánea:** Recurso del laboratorio que representa instrumentos de trabajo.
- ✓ **Objeto de Cristalería:** Recurso del laboratorio que representa objetos de vidrio.
- ✓ **Material Biológico:** Representa cualquier parte viva o muerta proveniente de un organismo.
- ✓ **Reactivo Químico:** Representa una sustancia que se une a otros elementos y permite la preparación de una muestra para su posterior diagnóstico o estudio.

2.1.2 -Diagrama de clases del Modelo de Dominio.

El Modelo del Dominio se representa a través de un diagrama de clases en el que se muestran los conceptos u objetos del dominio del problema, las asociaciones entre ellas y los atributos que poseen. Su objetivo principal es brindar una mayor comprensión de los eventos que suceden en el entorno que se desea informatizar. A continuación se representan los conceptos que anteriormente fueron descritos y la relación entre ellos. (Ver figura 2.1)

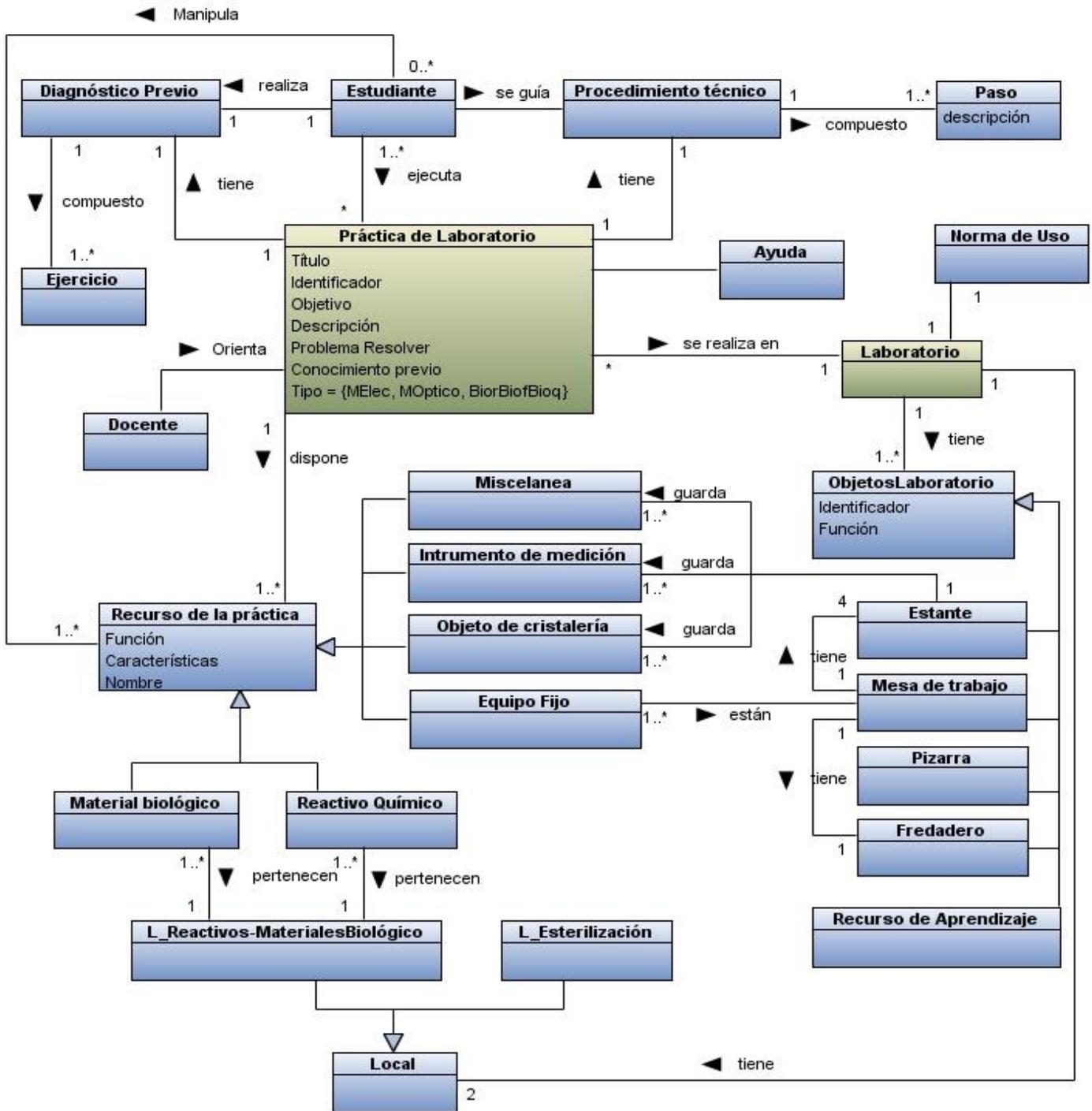


Figura 2.1. Representación del Modelo de Dominio que captura los conceptos más importantes en el contexto del sistema.

2.2 - Descripción del sistema propuesto

Para dar solución a los objetivos anteriormente planteados, como resultado del análisis se identificaron dos actores fundamentales (Docente y Estudiante). El docente es el responsable de asignar las prácticas de laboratorio a los estudiantes y el estudiante es el encargado de realizarlas, es la persona a quien va dirigido el laboratorio virtual de Biología. El estudiante previamente antes de acceder debe realizar un ejercicio de diagnóstico evaluativo, el cual ha sido asignado por el docente, a partir de la realización del ejercicio evaluativo. Se le asignarán al estudiante las prácticas de laboratorio a realizar que serán mostradas al usuario a través de una vista que contiene un listado con todas las prácticas de laboratorio asignadas una vez autenticado el usuario. Ambos actores realizan operaciones similares, razón por lo que se generalizan en usuario del sistema. El laboratorio de Biología consta de tres vistas principales para la realización de las prácticas. Además ambas vistas tienen asociadas dos opciones las cuales se encuentran disponibles en todo momento, tales son los procedimientos técnicos de la práctica de laboratorio asignada y una ayuda que contiene toda la información referente a la práctica a realizar.

La vista principal del laboratorio, es el ambiente donde se efectúa la práctica seleccionada. En esta vista se seleccionan los instrumentos de trabajo y se realizan los procedimientos técnicos del experimento, la misma está estrechamente relacionada con las demás vistas ya que permite el acceso a las mismas. Entre las asociadas, la vista concerniente al área de Reactivos y Materiales Biológicos, en este espacio se encuentran los reactivos y materiales biológicos, los cuales se seleccionan y se utilizan posteriormente en la vista principal del laboratorio. Además está la vista correspondiente al área de esterilización, en este espacio se simula todo el procedimiento de esterilización de los utensilios de trabajo utilizados en la práctica de laboratorio. El sistema propuesto de manera general describe los procedimientos comunes que se realizan en una práctica de laboratorio de Biología, centrándose específicamente en la descripción de dos funcionalidades de una de las prácticas de laboratorio seleccionada.

2.3 -Descripción de la funcionalidad

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requerimientos. Estos cumplen un papel primordial en el proceso de desarrollo del software, ya que marcan el punto de partida para actividades como la planeación, contribuyen a la identificación de las funcionalidades, el comportamiento de entrada y salida del sistema. Los requerimientos generan especificaciones correctas que describen con claridad, sin

ambigüedades y en forma consistente las características o cualidades que debe poseer la futura aplicación. Se pueden clasificar en: funcionales y no funcionales.

2.3.1 - Requerimientos Funcionales

Los Requerimientos Funcionales de un sistema especifican acciones que este debe ser capaz de realizar, sin considerar restricciones físicas. El sistema propuesto debe ser capaz de:

- R1. Mostrar escenario de selección de las Prácticas de Laboratorio.
- R2. Mostrar un listado de las prácticas de laboratorio clasificadas por tipo de prácticas.
- R3. Permitir seleccionar una práctica de Laboratorio.
- R4. Permitir acceder al laboratorio.
- R5. Mostrar las indicaciones de la práctica (objetivos, utensilios procedimientos técnicos, etc.) a seguir por el estudiante para la realización de cada práctica de laboratorio.
- R6. Mostrar equipo Fijo en la mesa de trabajo de realización de las prácticas (microscopio óptico y estereoscópico, balanza, espectrofotómetro, centrífuga, pH metro, micrótopo de mesa y baño de maría).
- R7. Mostrar los recursos necesarios para la realización de la práctica de laboratorio por estantes según tipo de práctica a ejecutarse.
 - R7.1 Mostrar recursos del estante de Misceláneas.
 - R7.2 Mostrar recursos del estante de instrumentos de medición.
 - R7.3 Mostrar recursos del estante de Cristalería.
 - R7.4 Mostrar recursos del estante de Equipos no fijos.
 - R7.5 Seleccionar material biológico del cuarto de reactivos y materiales biológicos.
- R8. Permitir la realización de los procedimientos de las prácticas.
 - R8.1 Permitir verter sustancia.
 - R8.2 Permitir manipular erlenmeyer.
- R9. Mostrar guía de procedimientos técnicos según la práctica seleccionada.

R10. Mostrar información con normas y cuidados a tener en el laboratorio de Biología.

R11. Permitir esterilizar los instrumentos utilizados en la práctica.

R12. Mostrar una ayuda por instrumentos y materiales de cada práctica a realizar.

R13. Permitir salir de la aplicación.

2.3.2 -Requerimientos No Funcionales.

RNF1. Apariencia o interfaz externa.

- ✓ La aplicación se ejecutará dentro de la plataforma en un navegador web.
- ✓ La aplicación tendrá una interfaz sencilla, intuitiva y amigable para sus usuarios.
- ✓ Se deben usar colores llamativos, ordenados en una gama coherente.
- ✓ El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.
- ✓ Las opciones de servicios además de tener su icono identificador tendrá el texto que muestre la opción en cuestión con su función descrita en una palabra para un reconocimiento rápido por el usuario y estos se mostraran siempre visibles.

RNF2. Soporte.

- ✓ Tarjeta de vídeo integrada de 128 MB (como mínimo)
- ✓ Tarjeta de sonido digital integrada de 128 BIT.

RNF3. Portabilidad

- ✓ Multiplataforma. El producto podrá ser ejecutado bajo los sistemas operativos XP y Linux en sus distintas versiones.

RNF4. Ayuda

- ✓ El producto contará con una ayuda contextualizada por instrumentos y según la práctica asignada.

RNF5. Diseño.

- ✓ Para el diseño se requiere del uso de Visual Paradigm, pues la metodología usada es RUP a través del lenguaje ApEM-L.

- ✓ El lenguaje de programación en la que se basa el diseño es Action Script 3.0.
- ✓ El framework utilizado para el desarrollo de la aplicación es el siguiente: Flex SDK 4.0.

RNF6. Hardware.

Los requerimientos que a continuación se muestran son los pertenecientes a la configuración mínima que debe tener la Terminal:

- ✓ Pentium 233 MHz (recomendado: Pentium 500MHz o mejor)
- ✓ 64 MB RAM (recomendado: 128 MB RAM o mejor)
- ✓ 1 GB espacio en Disco Duro
- ✓ Lector CD
- ✓ Dispositivo de audio
- ✓ Soporte de video que admita resolución de al menos 800x600 y 24 bits.
- ✓ Dispositivo de red de al menos 10 megabits.

RNF7. Software.

Estos productos se deben realizar en el sistema operativo Linux: Ubuntu 10.10 a propuesta del cliente.

RNF8. Legales, Derecho de Autor y otros

Una vez terminado el sistema debe ser sometido a una evaluación y certificación por parte del cliente del producto.

2.4 -Descripción de los actores que interactúan con el sistema.

Actor	Descripción
Estudiante	Persona que interactúa con el laboratorio, trabaja con la aplicación y a quien va dirigida la misma. Accede al laboratorio y realiza las actividades de experimentación asociadas a la práctica de laboratorio.
Docente	Orientador, controlador y evaluador de las actividades a desarrollar por el

	estudiante. Tiene la posibilidad de acceder al laboratorio y realizar las actividades de experimentación asociadas a la práctica de laboratorio.
Usuario	Es una generalización de los actores del sistema: Estudiante y Docente que responde a funcionalidades comunes en su interacción con el laboratorio.

2.5 -Vista de Gestión del Modelo y sus Subsistemas.

Con el surgimiento del artefacto Entorno de Presentación se define por ApEM-L el principio de organización del sistema representado a través de la Vista de Gestión del Modelo, el cual plantea que “*un subsistema de la aplicación puede estar compuesto por <n> Entornos de Presentación y a su vez un Entorno de Presentación puede ser una agregación de otros <n> Entornos de Presentación que contenga en su interior*” (1). Las relaciones entre los subsistemas van a ser representadas mediante relaciones de dependencia. Para la construcción del laboratorio virtual de Biología se han identificado cuatro subsistemas, definidos por las características y funcionalidades de los entornos que tiene asociado tal y como se presenta en la figura 2.2.

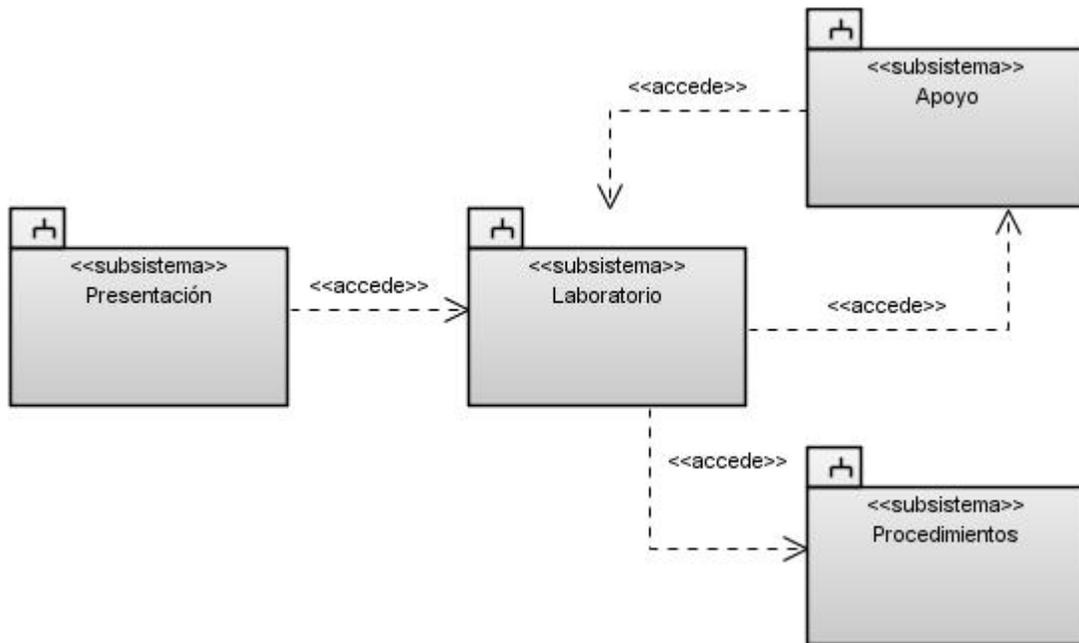


Figura 2.2 Representación de la Vista de Gestión del Modelo.

A continuación se exponen cada uno de los subsistemas y los entornos de presentación que lo componen, así como una breve explicación de la funcionalidad de estos.

Subsistema Presentación: Engloba todas las funcionalidades referente a la presentación del laboratorio antes de acceder a este, donde se muestra el listado de las prácticas de laboratorio que pueden ser realizadas por el estudiante una vez que la plataforma controla el cumplimiento satisfactorio del ejercicio diagnóstico por el estudiante. Está compuesto por el entorno donde se muestra el listado de prácticas y la forma de acceder al entorno del laboratorio virtual perteneciente al subsistema Laboratorio.

Entornos que contiene:

- ✓ Listado de prácticas.

Subsistema Laboratorio: Encierra todo el comportamiento interactivo de la práctica de laboratorio a realizar, así como la posibilidad de interactuar con las áreas de reactivos y materiales biológicos y esterilización. Está compuesto por los entornos que permiten el desarrollo de la práctica de laboratorio. Además se relaciona con el subsistema Apoyo y accede a los entornos del subsistema Procedimientos.

Entornos que contiene:

- ✓ Laboratorio Virtual de Biología.
- ✓ Indicaciones de la práctica.
- ✓ Área de materiales biológicos y reactivos.
- ✓ Área de esterilización.

Subsistema Procedimientos: Encierra todo el comportamiento interactivo de los procedimientos a realizar en la práctica de laboratorio. Está compuesto por los entornos que describen los procedimientos de manipulación con las sustancias y el recurso del Erlenmeyer.

Entorno que contiene:

- ✓ Verter sustancia.
- ✓ Manipular Erlenmeyer.

Subsistema Apoyo: Engloba todo el comportamiento de apoyo que permiten al usuario obtener información útil para su desenvolvimiento en la práctica a realizar y además permite la confirmación de la salida del sistema. Está compuesto por los entornos donde se muestra la ayuda, los procedimientos técnicos y la salida del sistema.

Entornos que contiene:

- ✓ Ayuda del laboratorio.
- ✓ Procedimientos técnicos.
- ✓ Confirmar salida.

2.6 -Descripción textual de los Entornos de Presentación.

La descripción textual de los Entornos de Presentación surge para una mejor descripción del contexto productivo del software educativo, incluyendo objetivos pedagógicos, elementos asociados a los entornos y medias a utilizar. Su objetivo principal es dejar claro en detalle todo el flujo de eventos así como todas las características esenciales y necesarias del entorno de presentación. A continuación se presenta la descripción textual del entorno de presentación correspondiente al Subsistema Laboratorio. (1)

✓ Subsistema Laboratorio

Tabla 2.1 Descripción textual del entorno de presentación “Laboratorio Virtual de Biología”.

Descripción Textual del entorno de presentación Laboratorio Virtual de Biología	
Actores del entorno	Usuario
Propósito	Permitir mostrar el laboratorio virtual con todos los objetos que recrean su espacio y realizar la práctica de laboratorio de Biología seleccionada por el usuario.
Objetivos pedagógicos²	Identificar los recursos necesarios para la realización de la práctica de laboratorio de Biología. Familiarizarse con los procedimientos técnicos para la realización de la práctica

² Parte de la descripción en la cual se mencionan los objetivos que desde el punto de vista pedagógico se logrará en el actor del sistema con la realización del entorno.

	<p>de laboratorio de Biología.</p> <p>Dominar las características técnicas esenciales para la realización de la práctica de laboratorio de Biología.</p>	
Resumen	<p>El entorno se inicia cuando el actor selecciona la opción “Acceder al Laboratorio”, el sistema muestra el laboratorio virtual de Biología permitiendo al usuario seleccionar los utensilios que necesita para realizar la práctica. Además permite al usuario colocar los diferentes utensilios en la mesa de trabajo de manera que pueda realizar el experimento finalizando el entorno con la visualización del experimento.</p>	
Entornos asociados	<p>“Indicaciones de la Práctica”, “Verter sustancia”, “Manipular Erlenmeyer”, “Confirmar salida”, “Área de esterilización”, “Área de reactivos y materiales biológicos”, “Procedimientos técnicos”, “Ayuda del laboratorio”.</p>	
Referencias	<p>R6,R7,R7.1,R7.2,R7.3,R7.4,R10</p>	
Precondiciones	<p>Haber seleccionado la opción “Acceder al laboratorio” del entorno de presentación “Listado de práctica”.</p>	
Poscondiciones	<p>Haber mostrado un mensaje de culminación de la práctica de laboratorio por el usuario.</p>	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	Elementos del entorno³
<p>1. El entorno se inicia cuando el actor elige la opción de “Acceder al laboratorio” en el entorno de Listado</p>	<p>2. El sistema muestra los elementos que conforman el ambiente del laboratorio tales como:</p> <ul style="list-style-type: none"> • La <i>imagen de fondo</i> que representa al laboratorio de Biología. • Una <i>pizarra</i> con indicaciones pedagógicas fundamentales de la práctica de laboratorio. 	<p>2.1 Se muestra un efecto de disolución de la imagen de fondo.</p> <p>2.2 La mesa de trabajo tiene en su interior un fregadero con una llave</p>

³ Parte de la descripción en la cual se describen los elementos de interacción que intervienen en el flujo de suceso del entorno, así como las características de diseño, propiedades, efectos que pudieran suceder.

<p>de prácticas.</p>	<p><u>(Ver entorno de presentación “Indicaciones de la práctica”)</u></p> <ul style="list-style-type: none"> • La <i>mesa de trabajo</i> para la realización del experimento con sus estantes: <ul style="list-style-type: none"> - Miscelánea. - Cristalería. - Equipos no fijos. - Instrumentos de medición. • Una <i>lámina</i> con las normas de uso del laboratorio. • Un <i>recuadro que representa el objeto de aprendizaje</i> relacionado con la práctica. • Los <i>equipos fijos</i> asociados a la práctica a realizar. • Un cesto de desechos. 	<p>de agua, solo se utiliza para verter agua.</p> <p>2.3 Los estantes están situados uno del lado del otro en la parte inferior de la mesa de trabajo.</p> <p>2.4 Al situarse encima de los estantes se iluminan estos.</p> <p>2.5 Al situarse encima de la lámina que representa las normas de uso del laboratorio y del recuadro que representa el objeto de aprendizaje se produce un efecto de iluminación.</p> <p>2.6 Los equipos fijos se sitúan encima de la mesa de trabajo en la porción derecha de la misma.</p> <p>2.7 Al verter cualquier objeto en el cesto de desecho se produce un sonido y este se tambalea durante dos</p>
----------------------	--	---

		segundos.
	<p>3. El sistema muestra las opciones que representan las secciones de apoyo a la realización de la práctica de laboratorio:</p> <ul style="list-style-type: none"> • Opción “Procedimientos técnicos”. • Opción “Ayuda”. 	<p>3.1 Se ilumina la opción “Procedimientos Técnicos” al pasar el mouse encima.</p> <p>3.2 Se ilumina la opción “Ayuda” al pasar el mouse encima.</p>
	<p>4. El sistema muestra las opciones para acceder a las otras áreas del laboratorio:</p> <ul style="list-style-type: none"> • Área de “Reactivos y materiales biológicos”. • Área de “Esterilización”. 	<p>4.1 Ambas opciones representan una puerta y se sitúan en la porción derecha del laboratorio virtual.</p> <p>4.2 Ambas opciones se iluminan al pasar el mouse encima.</p>
	<p>5. Se muestra la opción para la salida del laboratorio.</p>	
	<p>6. El sistema brinda la posibilidad de realizar la acción de:</p> <ul style="list-style-type: none"> • Seleccionar un recurso de los estantes de “Miscelánea”, “Cristalería”, “Instrumentos de medición”, “Equipos no fijos”. 	
<p>7. *Selecciona el estante</p>	<p>8. Muestra la animación de la apertura de la puerta del estante seleccionado.</p>	<p>8.1 Se reproduce un sonido simulando la apertura de las puertas del estante.</p>

<p>9. *Elige el recurso de trabajo del estante que necesita para la realización de la práctica.</p>	<p>10. Verifica si el recurso de trabajo elegido es el correcto según la práctica a realizar.</p>	
	<p>11. Verifica que no exista algún otro recurso del estante que depende del recurso elegido.</p>	
	<p>12. Permite llevar el recurso elegido hacia la mesa de trabajo.</p>	<p>12.1 Se visualizan en la mesa de trabajo áreas válidas para situar el recurso elegido. Las áreas son ovaladas de color blanco con un tono transparente.</p>
<p>13. *Sitúa el recurso en las áreas validas de la mesa de trabajo.</p>	<p>14. Muestra en la mesa, el recurso que ha sido seleccionado por el usuario.</p>	<p>14.1 Se reproduce un sonido sino se posiciona correctamente el utensilio encima de la mesa de trabajo.</p>
	<p>15. El sistema brinda la posibilidad de realizar las acciones de:</p> <ul style="list-style-type: none"> • Ejecutar los procedimientos de la práctica. • Elegir un nuevo recurso de trabajo del estante. (<i>Ver Flujo alterno 15.a</i>) • Retirar el recurso elegido de la mesa de trabajo. (<i>Ver Flujo alterno 15.b</i>) • Elegir una nueva acción en el laboratorio (<i>Ver</i> 	

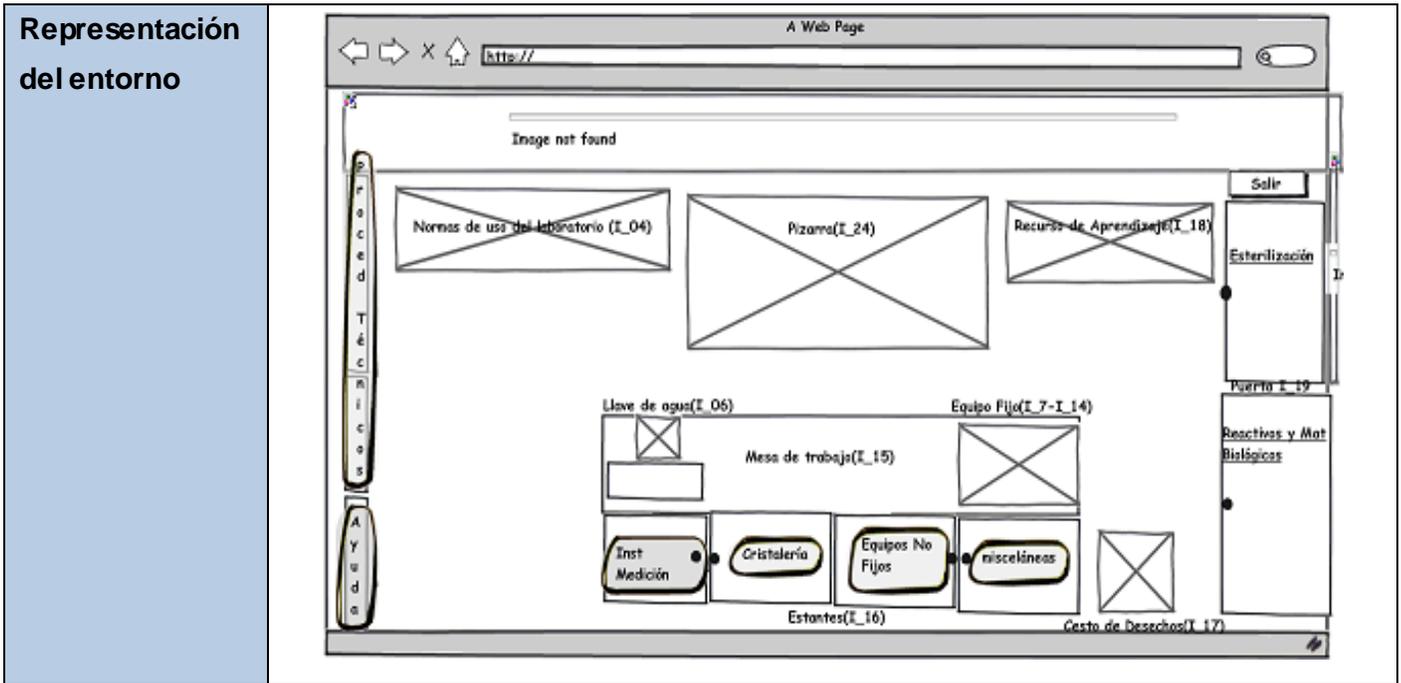
	<u>flujo alterno 15.c)</u>	
16. Decide ejecutar los procedimientos de la práctica.	17. El sistema verifica si están todos los recursos necesarios para ejecutar los procedimientos de la práctica.	
	18. El sistema brinda la posibilidad de realizar los siguientes procedimientos: <ul style="list-style-type: none"> • Procedimiento de verter sustancia. <u>(Ver entorno de presentación “Verter sustancia”)</u> • Procedimiento de manipulación del objeto de cristalería Erlenmeyer. <u>(Ver entorno de presentación “Manipular Erlenmeyer”)</u> 	
	19. Se culmina las funcionalidades asociadas al entorno de presentación.	
Curso alterno de los eventos 6.a “Otra acción”		
Acción	Curso alterno	
	6. a.1 El sistema brinda la posibilidad de realizar la acción de: <ul style="list-style-type: none"> • Seleccionar un recurso del local de reactivos y materiales biológicos. <u>(Ver entorno de presentación “Área de reactivos y materiales biológicos”).</u> 	
	6. a.2 Se retoma a la acción 6 del flujo básico del entorno de presentación.	
Curso alterno de los eventos 10.a “Recurso erróneo”		
Acción	Curso alterno	
	10.a.1 En caso de que el recurso de trabajo elegido del estante de la Miscelánea no sea el correcto el sistema muestra un cartel indicando que el utensilio seleccionado no puede ser utilizado para la realización de la actividad	
	10. a.2 Se retoma a la acción 9 del flujo básico del entorno de presentación.	
Curso alterno de los eventos 11.a “Recurso dependiente”		

Acción	Curso alternativo		
	11. a.1 En caso de que el recurso exista pues el sistema muestra un cartel indicando que se necesita en la mesa de trabajo otro recurso que depende del elegido (T_02).		
	11. a.2 Se retoma a la acción 9 del flujo básico del entorno.		
Curso alternativo de los eventos 15.a “Nuevo recurso”			
Acción	Curso alternativo		
	15. a.1 Se retorna a la acción 9 del flujo básico del entorno de presentación.		
Curso alternativo de los eventos 15.b “Retirar recurso”.			
Acción	Curso alternativo		
	15. b.1 El usuario selecciona el recurso a retirar de la mesa de trabajo y lo coloca en el estante correspondiente.		
	15. b.2 Se retoma a la acción 6 del flujo básico del entorno de presentación.		
Curso alternativo de los eventos 15.c “Nuevo acción en el laboratorio”.			
Acción	Curso alternativo		
	15. c.1 Se retoma a la acción 6 del flujo básico del entorno de presentación.		
Curso Alterno de los eventos 17.a “Recursos incompletos”.			
Acción	Curso alternativo		
	17. a.1 En caso de que no estén todos los recursos necesarios para ejecutar los procedimientos de la práctica el sistema muestra un cartel indicándole al usuario la necesidad de otros recursos.		
	17. a.2 Se retoma a la acción 6 del flujo básico del entorno de presentación.		
Prioridad	Crítica		
Mejoras	-		
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_04	Imagen que representa una lámina con las normas	En construcción

		de uso del laboratorio.	
	I_05	Imagen que representa un librero.	En construcción
	I_06	Imagen que representa el fregadero el cual contiene una llave de agua.	En construcción
	I_07	Imagen que representa el microscopio óptico	En construcción
	I_08	Imagen que representa el microscopio estereoscópico	En construcción
	I_09	Imagen que representa la balanza.	En construcción
	I_10	Imagen que representa el espectrofotómetro	En construcción
	I_11	Imagen que representa la centrífuga.	En construcción
	I_12	Imagen que representa el pH metro.	En construcción
	I_13	Imagen que representa el micrótopo de mesa	En construcción
	I_14	Imagen que representa el baño de maría	En construcción
	I_15	Imagen que representa la mesa de trabajo	En construcción
	I_16	Imagen que representa los estantes de la mesa de trabajo.	En construcción
	I_17	Imagen a que representa el cesto de desechos.	En construcción
	I_18	Imagen asociada al recurso de aprendizaje.	En construcción
	I_19	Imagen que representa la puerta	En construcción
	I_20	Imagen que representa el botón de salir del sistema.	En construcción
	I_23	Imagen que representa el fondo del laboratorio.	En construcción
Animación	A_01	Animación que representa la apertura del estante.	En construcción

Sonido	S_02	Sonido que representa la caída de instrumento en la mesa de trabajo.	En construcción
	S_03	Sonido que representa la apertura de las puertas del estante de los instrumentos de trabajo.	En construcción
	S_04	Sonido que representa la apertura de la puerta.	En construcción
	S_05	Sonido que representa verter un desecho en cesto.	En construcción
Texto	T_02	Texto que indica el mensaje de error de utensilio. <i>“Estimado estudiante el utensilio seleccionado no pertenece a esta práctica”.</i>	Existente.
	T_03	Texto que indica que el utensilio seleccionado depende de otro utensilio. <i>“Estimado estudiante se necesita otro utensilio”.</i>	Existente.
	T_04	Texto que indica el nombre del utensilio.	En construcción
Video	-	-	-
Reglas Pedagógicas⁴	-		

⁴ Parte de la descripción en la cual se reflejan los elementos que no pueden ser expresados en el cuerpo del entorno (reglas pedagógicas o instructivas) a tomar en consideración para la implementación del entorno.



Las descripciones textuales del resto de los entornos de presentación pueden ser consultadas en los Anexos 1-9.

2.7 -Diagramas de Estructura de Presentación

Los Diagramas de Estructura de Presentación (DEP) permiten representar la estructura que tendrán las interfaces de usuario, estableciendo una organización lógica de los elementos conformantes de las mismas y dejándoles a los diseñadores gráficos la función de decidir dónde y cómo serán en términos visuales dichos elementos.

Para la mejor estructuración del modelo ApEM-L define dos nuevos tipos de clases: la *clase Estáticos* y la *clase Interacción*, las cuales seccionarán los elementos que cumplan con cada una de las características que denotan los propios nombres. La clase estáticos agrupará los componentes que solo tienen como función visualizar información, pero que no permiten interacción con el usuario. Todo lo contrario con los agrupados bajo la clase interacción, los cuales serán los elementos de la vista que permiten interacción del usuario con el sistema informático modelado. Para cada uno de los entornos de presentación se

realiza DEP. A continuación se muestra el DEP del entorno de presentación principal del laboratorio virtual de Biología.

✓ **Subsistema Laboratorio**

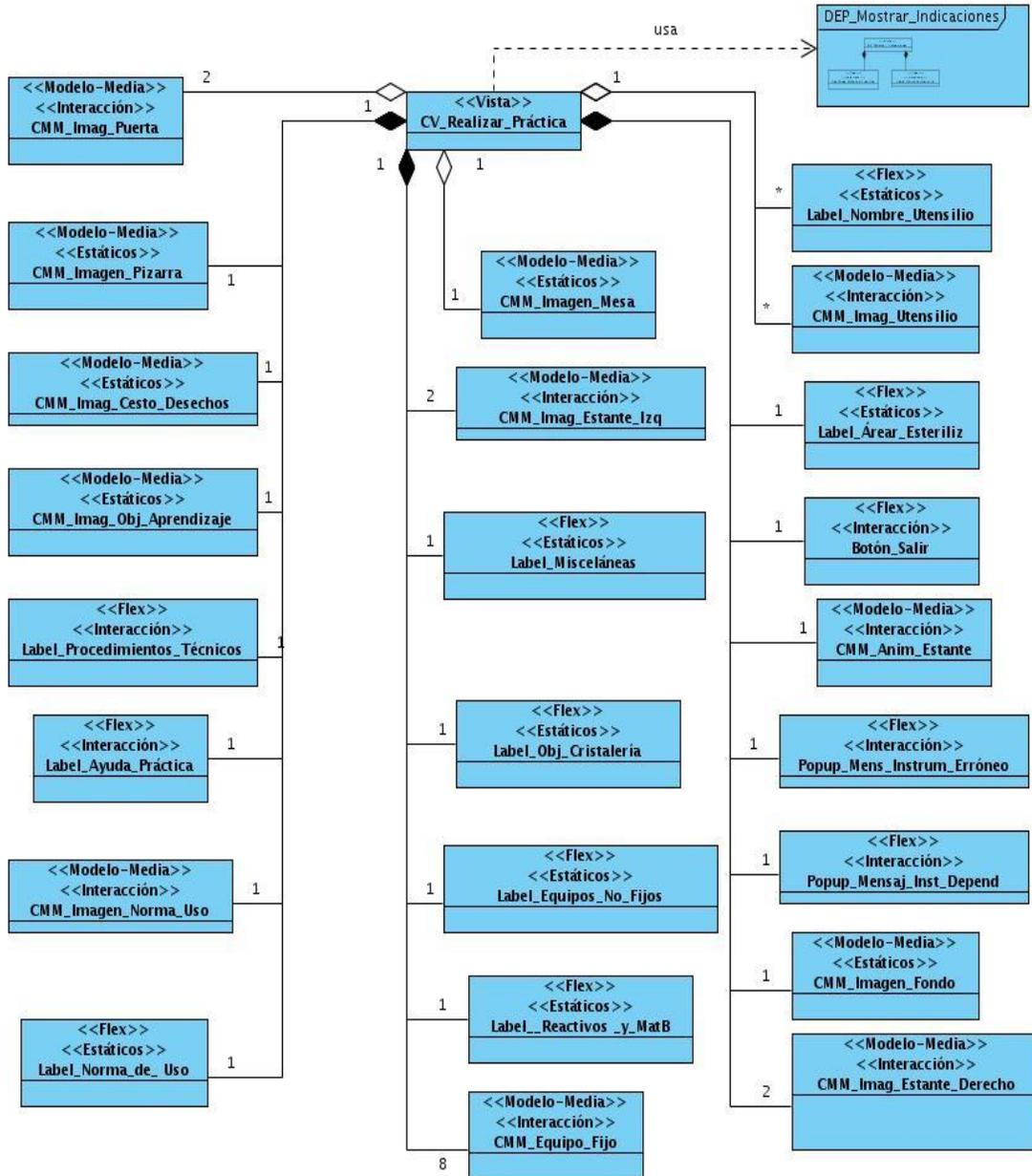


Figura 2.3 DEP Entorno de Presentación “Laboratorio Virtual de Biología”.

Los Diagramas de Estructura de Presentación de los demás entornos de presentación pueden ser consultados en los Anexos 10-19.

2.8 -Diagramas de Estructura de Navegación

Debido a la ineficiencia del lenguaje base UML para modelar la navegación en el software educativo se ha definido el Diagrama de Estructura de Navegación (DEN) como un artefacto de ApEM-L. El DEN representa la estructura y organización de los elementos de navegación de sistemas incorporando nuevos estereotipos restrictivos que facilitan la comprensión del modelo como son: *la clase botón (<<botón>>)*, *la clase índice (<<índice>>)*, *la clase consulta (<<consulta>>)* y *la clase menú (<<menú>>)*, además de utilizar las clases de los medios de comunicación que representan navegación como: modelo-entidad-media imagen y modelo-entidad-media texto. Estos estereotipos son elementos de composición de la clase vista que permiten el enlace con otras clases vistas del modelo.

La clase menú es el elemento de composición de una clase vista, desde donde se puede llegar a otras diferentes clases vistas con las cuales se conecta este menú, pues contiene una lista de las opciones de movimiento siguiente, pero no los valores de esas opciones, los cuales serán ofrecidos por el resto de los tipos de clases. La clase consulta es aquel tipo de clase que permite el enlace con una clase vista a y través de un valor directo (variable de consulta) asignado a la búsqueda para la visualización del elemento a mostrar. La clase índice es aquella clase que denota el valor de direccionamiento hacia una clase vista a partir de una opción determinada. Por último la clase botón es aquella clase que denota la existencia de un elemento interactivo del tipo botón para producir un camino en la navegación hacia una clase vista.

Es válido destacar en la representación de este artefacto la existencia de un nuevo estereotipo que representa quién inicializa la navegación en el sistema, ya que en versiones anteriores de ApEM-L no existía la manera de cómo representar el usuario que da comienzo a la navegación y a través del cual el entorno de presentación o elemento interactivo se realiza. De aquí que exista la necesidad de insertar en este diagrama el artefacto actor del sistema tal y como lo define y representa UML por ser un elemento fundamental en el inicio de la navegación del software. (1)

A continuación se muestra el DEN del laboratorio virtual de Biología.

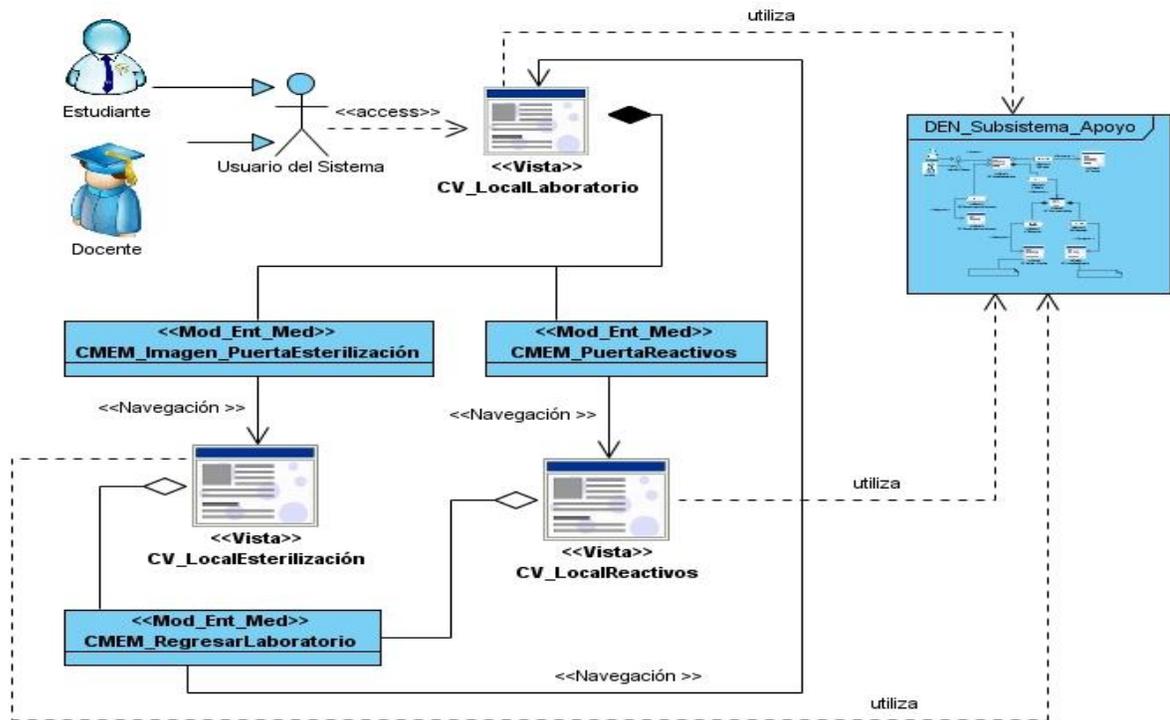


Figura 2.4 DEN Subsistema Laboratorio

Los DEN del resto de los subsistemas pueden ser consultados en los Anexos 20-21

Conclusiones parciales

En este capítulo se realizó la identificación de los actores del sistema y se describió la solución del sistema propuesto. Se identificaron y describieron los principales conceptos del entorno donde se usará el software, se realizó el modelamiento del dominio y se especificaron trece requerimientos funcionales y siete no funcionales que debe cumplir el software a desarrollar. Además se representó la Vista de Gestión del Modelo, compuesta por cinco subsistemas. También se hizo una descripción textual de uno de los principales Entornos de Presentación, se mostró el Diagrama de Estructura de Presentación asociado a este entorno y el Diagrama de Estructura de Navegación correspondiente al subsistema principal del laboratorio de Biología.



CAPÍTULO III: DISEÑO DE LA SOLUCIÓN PROPUESTA

Introducción

En este capítulo se realiza una descripción del patrón arquitectónico utilizado y de los patrones de diseño a utilizar. Además se presentan los artefactos fundamentales que guiarán el trabajo correspondiente a la implementación de la solución propuesta. Dentro de los que se muestran los diagramas de clases del diseño y los diagramas de secuencia, con ciertas modificaciones propuestas por ApEM-L. Además se presenta el Modelo-Entidad-Relación de las clases persistentes del sistema a construir.

3.1 - Patrón Arquitectónico y Patrones de Diseño Utilizados

El patrón arquitectónico utilizado para diseñar el laboratorio virtual de Biología es el MVC-E: una variante del MVC estándar, pero contextualizado al dominio particular del software educativo. Esta nueva extensión propone algunas transformaciones sobre la base del estudio comparativo de las responsabilidades de las clases del análisis definidas por el UML y las clases definidas por MVC. Tales modificaciones son: (26)

1. Mantener la clase Vista (Interfaz según UML) del patrón original con sus responsabilidades establecidas, es decir, recepción de las peticiones del usuario y mostrar las respuestas del sistema.
2. Mantener la separación que MVC establece entre las clases Modelo y Controlador, quitándole a la Clase Modelo la responsabilidad asociada con la gestión y almacenamiento de los datos persistentes en la aplicación.
3. Definir una nueva clase denominada Modelo-Entidad, que asuma la responsabilidad de la gestión y el almacenamiento de toda la información persistente de la aplicación, así como la comunicación con la base de datos del sistema en cuestión.

Todas estas modificaciones del MVC-E hicieron posible que se establecieran nuevos cambios semánticos y sintácticos al metamodelo base del UML, dando surgimiento a nuevos estereotipos que posibilitaran la representación de las nuevas características impuestas por dicho patrón. Este acontecimiento dio paso al

surgimiento de ApEM-L en el 2007 como extensión del UML y como tecnología de representación del software educativo.

ApEM-L 1.0 teniendo en cuenta las teorías del patrón MVC-E, define dos nuevos estereotipos: (1) <<Modelo-Entidad-Persistente>>, el cual según esta primera versión tiene como responsabilidad la gestión de la información persistente que antes sobrecargaba a la clase Modelo del patrón MVC original y (2) <<Modelo-Entidad-Media>> con la responsabilidad de agrupar las clases que identifican las medias y su árbol de jerarquía en la aplicación. Sin embargo a la hora de utilizar estos estereotipos en los diagramas de clases del diseño se evidencian algunas contradicciones desde el punto de vista sintáctico en correspondencia con la semántica original de estos:

1. En primer lugar la sintaxis del primero de los estereotipos tal y como está, le da un sentido de comprensión diferente al elemento del modelo que es estereotipado (en este caso el elemento *clase* del metamodelo⁵ del UML), ya que el mismo tiende a identificar a las clases persistentes de la aplicación y no a las clases que responden a la gestión y almacenamiento de la información persistente.
2. Por otro lado el utilizar el estereotipo con la sintaxis <<Modelo-Entidad-Media>> para la representación de las clases medias se está intuitivamente respondiendo a la semántica de la Modelo-Entidad definida por MVC-E, precisamente porque parte del nombre del estereotipo contiene esta terminología, elemento este que trae consigo contradicción en la comprensión de los modelos por los desarrolladores del software.

Teniendo en cuenta la situación planteada y tomando en consideración los principios teóricos del MVC-E, en conjunto con las teorías de los estereotipos (26) utilizadas por ApEM-L se proponen las siguientes modificaciones al lenguaje ApEM-L como propuesta a futuras versiones del lenguaje:

1. Realizar una variación de la sintaxis del estereotipo <<Modelo-Entidad-Persistente>> y dejarlo en <<**Modelo-Entidad**>> para darle el verdadero sentido de comprensión a las clases del diseño que responden a la gestión y el almacenamiento de la información persistente, tal y como lo define el patrón MVC-E.

⁵ Es la capa superior de la infraestructura de un lenguaje de modelado que representa los conceptos bases de este. Ejemplo el concepto clase, atributo, asociación, entre otros. (43)

2. Realizar una variación de la sintaxis del estereotipo <<Modelo-Entidad-Media>> y dejarlo en <<Modelo-Media>> para darle el verdadero sentido de comprensión a las clases del diseño que identifican las medias y su árbol de jerarquía en la aplicación ya que las mismas son clases Modelo en correspondencia con el patrón MVC original.

Por otra parte dentro de los patrones de diseño de software utilizados se empleó el Experto, este permitió que se asignaran responsabilidades a las clases que contaran con la información necesaria para cumplir tal responsabilidad. Se evidencia su aplicación en la clase *Práctica*, que como es la que cuenta con toda la información referente a las prácticas de laboratorio pues es la responsable de hacer que de ella se obtengan el objetivo, la descripción y los utensilios que utiliza la práctica de laboratorio. Otro patrón de diseño utilizado fue el Creador con el que se pudo determinar qué clase sería la responsable de crear alguna instancia de otra clase. Se aplica por ejemplo en la clase *Práctica*, la que es responsable de crear una instancia de la clase *Utensilios*, debido a que un utensilio será utilizado al realizar una determinada práctica de laboratorio.

3.2 - Modelo del Diseño

El modelo de diseño es un modelo de objetos que se centra en cómo los requerimientos funcionales y no funcionales, junto con otras restricciones del entorno de implementación, tienen impacto en el sistema que se desarrolla. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Entre sus principales propósitos están:

- ✓ Adquirir una comprensión de los aspectos relacionados con los requerimientos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- ✓ Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requerimientos o subsistemas individuales, interfaces y clases.
- ✓ Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación.

3.2.1 - Diagrama de Clases

El diagrama de clases es el modelado de determinados conceptos en forma de clases que describen un conjunto de objetos que almacenan información y se comunican para implementar su comportamiento. La información almacenada se representa como atributos de estas clases y las operaciones a través de los métodos de las mismas. Los diagramas de clase son un tipo de diagrama estático que constituye el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido. Son los diagramas más comunes en el modelado de sistemas orientados a objetos.

ApEM-L propone algunas modificaciones para este tipo de diagrama. La primera de estas modificaciones está relacionada con la organización del diagrama en dos grandes zonas. Por un lado la zona de la izquierda donde se representan las clases que identifican al árbol jerárquico de los productos medias de la aplicación y por otro lado la zona de la derecha del diagrama las clases que representan y controlan la lógica del negocio de la aplicación propiamente dicha. Esta zona de la derecha vuelve a subdividirse en cuatro zonas. La primera dedicada a las clases vista, la contigua a esta y en el extremo superior derecho dedicada a las clases controladoras, inmediatamente debajo de esta sección, la destinada a las clases modelo, quedando en banda superior derecha las clases modelo entidad para el tratamiento de la información persistente de la aplicación; y en la zona central aledañas a las clase vista las clases correspondientes al Lenguaje de Alto Nivel (HLL) con el que se programe (30), que en el caso del laboratorio sería las clases pertenecientes al Entorno de Desarrollo Integrado (IDE en sus siglas en inglés) Flex SDK 4.0

Este tipo de diagrama se representa para cada una de los entornos de presentación del sistema a desarrollar sobre la base de que sus elementos responden a una determinada funcionalidad del sistema. Además se aplican las modificaciones presentadas como propuesta en el epígrafe 3.1. Seguidamente se muestra el diagrama de clases perteneciente al Subsistema Laboratorio como realización del entorno de presentación más importante del sistema a construir.

Los diagramas de clases del resto de los Entornos de Presentación pueden ser consultados en los Anexos 22-30.

3.2.2 -Diagramas de Secuencia

Un diagrama de secuencia es un diagrama que representa el comportamiento del sistema y es una especialización de los diagramas de interacción que define el UML, pues como define la semántica de UML, representa la manera en la que los objetos de la aplicación intercambian mensajes para darle cumplimiento a sus responsabilidades.

ApEM-L no modifica la semántica del lenguaje base de dicho diagrama sino que se extiende esta al incorporar el estereotipo restrictivo asociado al tiempo y su representación siempre que sea necesario enriqueciendo la descripción funcional de este tipo de aplicaciones y por la importancia del control de esta variable en las aplicaciones educativas y en especial en el trabajo con los medios continuas de comunicación como sonido, video y animaciones. (30)

Este tipo de diagrama se representa para cada una de los entornos de presentación del producto de software. Seguidamente se muestra el diagrama de secuencia del entorno de presentación más importante del software a construir perteneciente al subsistema Laboratorio.

✓ Subsistema Laboratorio

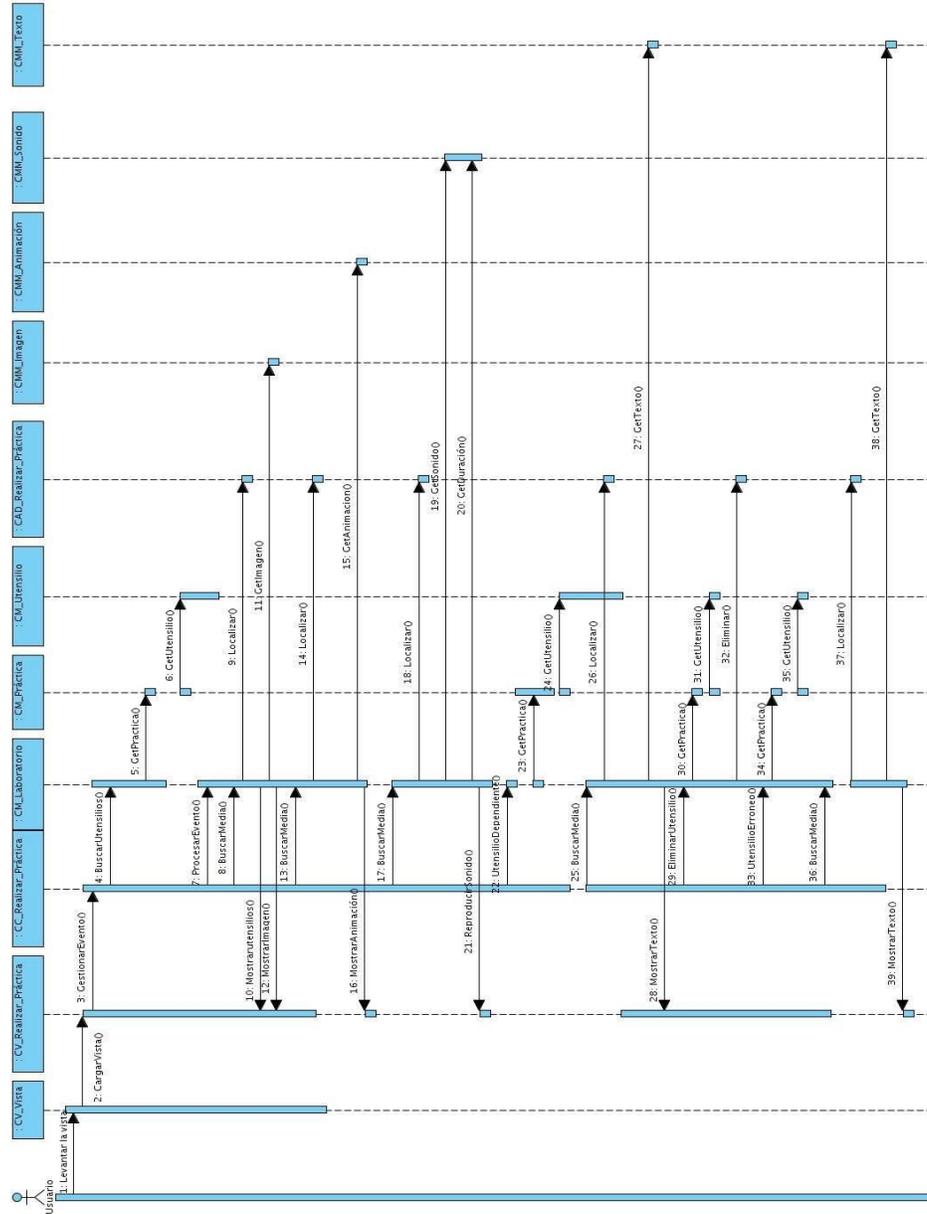


Figura 3.2 Diagrama de secuencia del Entorno de Presentación “Laboratorio Virtual de Biología”.

Los diagramas de secuencia del resto de los Entornos de Presentación pueden ser consultados en los Anexos 31-39.

3.3 - Diseño de la Base de Datos

La base de datos (BD) constituye un pilar importante dentro de la aplicación que se desarrolla ya que es la que mantiene todos los datos del proceso completo. A continuación se muestran las clases persistentes identificadas para la construcción del laboratorio de Biología.

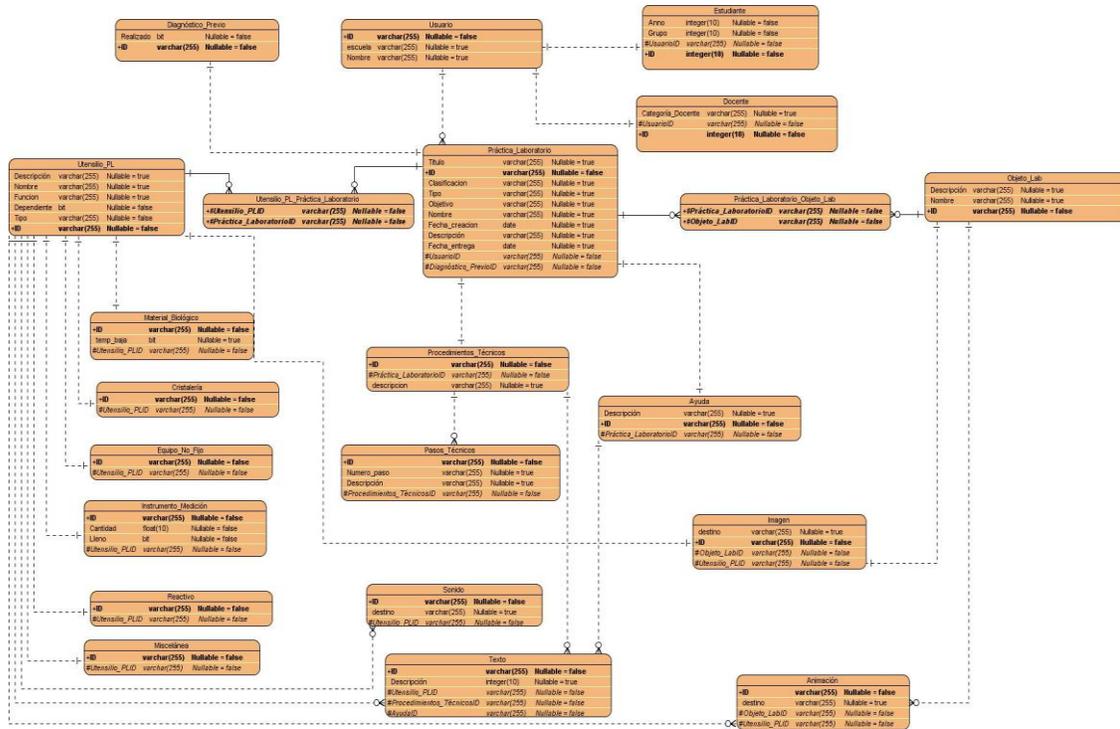


Figura 3.3 Diseño de la Base de Datos

3.3.1 –Descripción de las tablas

Las descripciones textuales de las tablas de la Base de Datos pueden ser consultadas en los Anexos 40-61

Conclusiones parciales

En este capítulo se hizo una descripción del patrón arquitectónico y de los patrones de diseño utilizados para el diseño del laboratorio virtual de Biología. Se obtuvieron los artefactos fundamentales correspondientes al diseño de la solución propuesta, se representó el diagrama de clases del diseño de y el diagrama de secuencia correspondiente al entorno principal del laboratorio de Biología. Finalmente se presentó la propuesta del diseño de la base de datos.

CONCLUSIONES GENERALES

Con la realización del presente trabajo de diploma, se obtuvo la documentación ingenieril que permite el análisis y diseño para el desarrollo de un laboratorio virtual de Biología, para la Plataforma de Aprendizaje ZERA.

El estudio del estado del arte realizado en los laboratorios virtuales de Biología, permitió la selección de las herramientas y artefactos necesarios para el desarrollo de la propuesta de solución. Se dejó constancia de un análisis del estado del arte que permitió realizar un levantamiento bibliográfico del objeto de investigación y de las metodologías de desarrollo y lenguajes de notación utilizados en el proceso de desarrollo del software especialmente en los utilizados para aplicaciones educativas.

La selección de los requisitos funcionales y no funcionales propuestos permitió realizar el análisis y diseño para el desarrollo del objetivo general de la investigación. La metodología RUP, empleada a partir del estudio realizado, permitió la obtención de los modelos y diagramas que modelan los aspectos relacionados a la navegabilidad, interactividad y funcionalidad consistentemente reflejada en la documentación obtenida.

RECOMENDACIONES

A partir de los resultados y beneficios obtenidos de la presente investigación, se realizan las siguientes recomendaciones:

1. Tener en cuenta el estudio realizado para la implementación y prueba del laboratorio virtual de Biología a partir del análisis y diseño realizado.
2. Continuar con el estudio de los laboratorios virtuales, sobre la base de los hiperentornos de aprendizaje para plataformas educativas, con vistas a lograr una mayor experiencia para las futuras mejoras a la propuesta.
3. Utilizar la documentación ingenieril obtenida como memoria para el análisis y diseño de laboratorios virtuales que respondan a otras disciplinas de la plataforma ZERA.
4. Añadir en próximas iteraciones del Laboratorio Virtual de Biología nuevas prestaciones que respondan a:
 - ✓ Procesos que respondan a otras prácticas de laboratorio. (Ej. Manipular microscopio electrónico, entre otros)
 - ✓ La gestión de la traza del estudiante
 - ✓ La gestión de evidencias de los estudiantes una vez realizada la práctica de laboratorio.
 - ✓ Realización de salvas automáticas durante la ejecución de la práctica.
5. Incorporar en ApEM-L el uso de estereotipos para lograr mayor comprensión de la modelación con este lenguaje.

REFERENCIAS BIBLIOGRÁFICAS

1. **Ciudad, Febe Angel y Herrera, Yosnel.** DISEÑO DE APLICACIONES EDUCATIVAS MULTIMEDIAS UTILIZANDO UNA NUEVA VERSIÓN DEL LENGUAJE DE MODELACIÓN ApEM – L. *Convención y Feria Internacional Informática Habana.* [En línea] Ministerio de la Informática y las Comunicaciones., Febrero de 2009. [Citado el: 3 de Diciembre de 2010.] <http://www.informaticahabana.cu/ponencia&trid=810>.
2. **Sánchez, Dr.Jaime.** *Construyendo y Aprendiendo con el Computador.* Universidad de Chile,Santiago de Chile : s.n., 1999.
3. **W. W., Lee y D. L., OWENS.** *Multimedia based instruction.* Massachussets : Jossey - Bass, 2000.
4. **Curbelo, Humberto y al., et.** *LA CLASE DE SOFTWARE EDUCATIVO EN LA ENSEÑANZA SECUNDARIA BASICA.* [CD-ROM] Ciudad de la Habana : Memorias de Pedagogía 2007, 2007.
5. **Rosales Velazquez, Norlen.** *"SOFTWARE EDUCATIVO PICO MESTRO".* La Habana : Universidad de las Ciencias Informáticas(UCI), 2009.
6. **Suárez Yanes, Diony y Quesada Costales, Michael.** *"Documentación Ingenieril del Software Educativo Jugar y Aprender."* La Habana : Universidad de las Ciencias Informáticas(UCI), 2008.
7. **Guerra Tamayo, Yaimara.** *InfoHis: Software Educativo para el estudio de la.* La Habana : Universidad de las Ciencias Informáticas(UCI), 2008.
8. **Zayas, Álvarez.** *Pedagogía como ciencia.* La Habana : Editoria Felix Varela, 1998.
9. **Babanski Yu, K.** *Optimización del proceso docente.* La Habana : Editorial Pueblo y Educación, 2003.
10. **Balboa, Roberto.** *Newton_T: un sistema de producción de entrenadores y tutores inteligentes.Memorias del Congreso iberoamericano de Informática Educativa.Tomo III.* Junio 2004.
11. **González Castro, Vicente.** *Teoría y Práctica de los medios de enseñanza.* La Habana : Editorial Pueblo y Educación, 1986.
12. **Coloma Rodríguez, M. Sc. Orestes.** *HIPERENTORNO DE APRENDIZAJE "EUREKA": UN SOFTWARE EDUCATIVO PARA LA ENSEÑANZA DE LA MATEMÁTICA .*
13. **Autores, Colectivo de.** *MINED. Dirección Computacional. . s.l. : 2003, La Habana.*
14. **Covadonga, A.A.** Entornos Formativos en el ciberespacio:las plataformas educativas. [En línea] [Citado el: 12 de diciembre de 2010.] <http://web.ebscohost.com/>.
15. **Agudelo, M.M.** Plataformas Educativas. [En línea] 2006. [Citado el: 15 de diciembre de 2010.]
16. **Monge Nágera, J.** *La evolución de los laboratorios virtuales durante una experiencia de seis años con estudiantes a distancia.* 1999.

17. **Wulf, W.** *El colaborador: Un nuevo enfoque hacia la investigación científica*. Seminario de la Fundación nacional para la Ciencia : s.n., 1989.
18. **Vary, J. P.** *Informe de la reunión de expertos sobre laboratorios virtuales* . s.l. : Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura., 1999.
19. **Muñoz Carballo, Liosdany y Rodríguez Escalona, Jorge E.** *Nodo Virtual de Procesos Versión 2.0*. La Habana : s.n., 2010.
20. **Woodfield, Brian F y Asplund, Matthew C.** *Corel ChemLab v1.0*. [CD-ROM] Canadá : Corel Corporation, 1996.
21. **Woodfield, Brian F. y Asplund, Matthew C.** *Virtual ChemLab: General Chemistry Laboratories*. [CD-ROM] Canada : Prentice Hall, 2001. 0-13-185751-7.
22. **Antón M.G.D., Pérez M.A.** *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica*. Caracas, Venezuela, Universidad Simón Bolívar(USB) : s.n., 2002.
23. **Jacobson I., G Booch.** *El proceso unificado de desarrollo de software*. Madrid : Addison Wesley, 2000. 84-7829-036-2.
24. **Martínez Pérez, Yaillet.** *Propuesta de proceso para el desarrollo de proyectos de SWE en la Universidad de las Ciencias Informáticas. [Tesis Maestría]*. Ciudad de la Habana : s.n., Nov 2009.
25. **Reynoso, C. B.** *Introducción a la Arquitectura de Software*. Universidad de Buenos Aires : s.n., 2004.
26. **Ciudad Ricardo, Febe Ángel.** *Utilización del Patrón Modelo-Vista Controlador(MVC) en el diseño de software educativos*. Departamento de la Especialidad Facultad 9, Universidad de las Ciencias Informáticas. Ciudad de la Habana : II Taller de Software Educativo e Hipermedia en la II Conferencia Científica UCIENCIA 2006, 2006. pág. 17, Artículo científico.
27. **Larman, Craig.** *UML y patrones. Introducción al análisis y diseño orientado a objetos*. . Mexico : Prentice Hall and Addison Wessley, 1999. 970-17-0261-1.
28. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico. Edición 6ta.* . s.l. : Mc Graw Hill, 2006. 970-10-5473-3.
29. **Orallo, E.H.** El lenguaje Unificado de Modelado(UML). [En línea] 2001. [Citado el: 13 de diciembre de 2010.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
30. **Ciudad Ricardo, Febe Ángel.** *ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*. Ciudad de la Habana, Universidad de las Ciencias Informáticas : s.n., 2007.

31. **Engels, Gregor y Sauer, Stefan.** Extending UML for Modeling of Multimedia Applications. [En línea] 2001. [Citado el: 23 de Enero de 2011.] <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>.
32. **Lorente A.E., Y.M. Pérez.** Planilla para el montaje dinámico de los productos de la colección Multisaber. [En línea] 2006. [Citado el: 9 de diciembre de 2010.] http://www.informaticahabana.com/evento_virtual/files/MUL067.pdf.
33. ActionScript. ActionScript Technology Center. [En línea] [Citado el: 15 de abril de 2011.] <http://www.adobe.com/devnet/actionscript.html>.
34. Flex SDK 4.0. Adobe.com. [En línea] [Citado el: 20 de mayo de 2011.] http://www.adobe.com/devnet/flex/articles/flex4sdk_whatsnew.html.
35. **Dwyer, T.** *Heuristic Strategies for Using Computers to Enrich Education.* 1974.
36. **Márquez, P.** *La Informática como medio didáctico: software educativo, posibilidades e integración curricular.* Murcia:DM : s.n., 2000.
37. **Botana, L.M.S.** *Evaluamos el Software Educativo?* 2007.
38. **Booch G., J. Rumbaugh.** El lenguaje unificado de modelado, Addison Wesley. [En línea] 2000. [Citado el: 2010 de diciembre de 15.] <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>.
39. **Galván, Lic. Liliana.** *E-PROCESO DE ENSEÑANZA-APRENDIZAJE.* s.l. : UPC.
40. **Torres, M.C., H.M. Espino.** *Colección Antonio Guiteras y Carlos Aponte: Seis libros para la formación de conciencia política.* Ciudad de la Habana, Universidad de las Ciencias Informáticas : s.n., 2008.
41. **Saúco, A. M.** *Análisis y diseño del juego interactivo Energía para Aprender.* Ciudad de la Habana, Universidad de las Ciencias Informáticas : s.n., 2008.
42. **Marqués Graells, P.** *Software Educativo. Guía de uso y metodología de diseño.* Barcelona : Editorial Estel, 1995 .
43. **OMG.** UML Infrastructure Specification, v 2.3. *Object Management Grupo.* [En línea] Mayo de 2010. [Citado el: 24 de Marzo de 2011.] <http://www.omg.org/spec/UML/2.3/Infrastructure/PS/.formal/2010-05-03>.
44. —. UML 2.0 OCL Specification. *Object Management Group.* [En línea] 1 de 5 de 2006. [Citado el: 27 de Enero de 2011.] <http://www.omg.org/spec/OCL/2.0/PDF/.ptc/03-10-14>.
45. **OMG Adopted Specification.** *Titulo UML 2.0 OCL Specification.* ptc/03-10-14.

BIBLIOGRAFÍA

- Marqués Graells, P.** *Software Educativo. Guía de uso y metodología de diseño.* Barcelona : Editorial Estel, 1995 .
- Sánchez, Dr. Jaime.** *Construyendo y Aprendiendo con el Computador.* Universidad de Chile, Santiago de Chile : s.n., 1999.
- Botana, L.M.S.** *Evaluamos el Software Educativo?* 2007.
- Autores, Colectivo de.** *Pedagogía.* La Habana : Editorial Pueblo y Educación, 2004. pp 182-301.
- Babanski Yu, K.** *Optimización del proceso docente.* La Habana : Editorial Pueblo y Educación, 2003.
- Balboa, Roberto.** *Newton_T: un sistema de producción de entrenadores y tutores inteligentes. Memorias del Congreso iberoamericano de Informática Educativa. Tomo III.* Junio 2004.
- González Castro, Vicente.** *Teoría y Práctica de los medios de enseñanza.* La Habana : Editorial Pueblo y Educación, 1986.
- Autores, Colectivo de.** *MINED. Dirección Computacional.* . s.l. : 2003, La Habana.
- Coloma Rodríguez, M. Sc. Orestes.** *HIPERENTORNO DE APRENDIZAJE "EUREKA": UN SOFTWARE EDUCATIVO PARA LA ENSEÑANZA DE LA MATEMÁTICA .*
- Covadonga, A.A.** Entornos Formativos en el ciberespacio: las plataformas educativas. [En línea] [Citado el: 12 de diciembre de 2010.] <http://web.ebscohost.com/>.
- Agudelo, M.M.** Plataformas Educativas. [En línea] 2006. [Citado el: 15 de diciembre de 2010.]
- Monge Nágera, J.** *La evolución de los laboratorios virtuales durante una experiencia de seis años con estudiantes a distancia.* 1999.
- Wulf, W.** *El colaborador: Un nuevo enfoque hacia la investigación científica.* Seminario de la Fundación nacional para la Ciencia : s.n., 1989.
- Antón M.G.D., Pérez M.A.** *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica.* Caracas, Venezuela, Universidad Simón Bolívar(USB) : s.n., 2002.
- Jacobson I., G. Booch.** *El proceso unificado de desarrollo de software.* La Habana : Felix Varela, 2004.
- Reynoso, C. B.** *Introducción a la Arquitectura de Software.* Universidad de Buenos Aires : s.n., 2004.
- Ciudad Ricardo, Febe Ángel.** *Utilización del Patrón Modelo-Vista Controlador(MVC) en el diseño de software educativos.* 2006.

- Ciudad Ricardo, Febe Ángel.** *ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI.* Ciudad de la Habana, Universidad de las Ciencias Informáticas : s.n., 2007.
- Orallo, E.H.** El lenguaje Unificado de Modelado(UML). [En línea] 2001. [Citado el: 13 de diciembre de 2010.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
- Lorente A.E., Y.M. Pérez.** Planilla para el montaje dinámico de los productos de la colección Multisaber. [En línea] 2006. [Citado el: 9 de diciembre de 2010.] http://www.informaticahabana.com/evento_virtual/files/MUL067.pdf.
- Torres, M.C.,H.M. Espino.** *Colección Antonio Guiteras y Carlos Aponte:Seis libros para la formación de conciencia política.* Ciudad de la Habana, Universidad de las Ciencias Informáticas : s.n., 2008.
- Jacobson I., G Booch.** *El proceso unificado de desarrollo de software.* La Habana : Editorial Felix varela, 2004.
- Booch G., J. Rumbaugh.** El lenguaje unificado de modelado, Addison Wesley. [En línea] 2000. [Citado el: 2010 de diciembre de 15.] <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>.
- Ciudad, Febe Angel y Herrera, Yosnel.** *DISEÑO DE APLICACIONES EDUCATIVAS MULTIMEDIAS UTILIZANDO UNA NUEVA VERSIÓN DEL LENGUAJE DE MODELACIÓN ApEM – L.* Ciudad de La Habana : s.n., 2009.
- Dwyer, T.** *Heuristic Strategies for Using Computers to Enrich Education.* 1974.
- Márquez, P.** *La Informática como medio didáctico:software educativo,posibilidades e integración curricular.* Murcia:DM : s.n., 2000.
- Galván, Lic. Liliana.** *E-PROCESO DE ENSEÑANZA-APRENDIZAJE.* s.l. : UPC.
- Saúco, A. M.** *Análisis y diseño del juego interactivo Energía para Aprender.* Ciudad de la Habana, Universidad de las Ciencias Informáticas : s.n., 2008.
- Larman, Craig.** *UML y Patrones.* México : s.n., 1999.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso Unificado de Desarrollo de Software.Manual de Referencia(traducción del original en inglés:The Unified Modeling Language.Reference Manual 1999).* Madrid : Addison-Wesley.1ra Edición, 2000.
- Engels, Gregor y Sauer, Stefan.** *Extending UML for Modeling of Multimedia Applications.* Alemania : s.n.
- OMG.** UML Infrastructure Specification, v 2.3. *Object Management Grupo.* [En línea] Mayo de 2010. [Citado el: 24 de Marzo de 2011.] <http://www.omg.org/spec/UML/2.3/Infrastructure/PS/.formal/2010-05-03>.

Ciudad Ricardo, Febe A. y Herrera Martínez, Yosnel. *ApEM-L 1.5: Una nueva versión del lenguaje de Modelado orientado a objeto para Aplicaciones Educativas y Multimedia.* La Habana : Revista Cubana de Ciencias Informáticas, Julio-Diciembre 2008. RCCI Vol 2, No. 3-4.

OMG Adopted Specification. *Titulo UML 2.0 OCL Specification.* ptc/03-10-14.