

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 5.**



**Marco de trabajo ingenieril para el desarrollo de
Laboratorios Virtuales con fines educativos.**



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

Autora: Yamilka de la Caridad Caceres Mayo.

Tutoras:

Ing. Susej Beovides Luis.

Ing. Yadira Ramírez Rodríguez.

La Habana, junio de 2013.

“Año 55 de la Revolución”

*“El éxito está más en función del sentido
común consistente que del genio”*

An Wang.

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma.

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

Yamilka de la Caridad Caceres Mayo.

Autora.

Ing. Susej Beovides Luis.

Tutora.

Ing. Yadira Ramírez Rodríguez.

Tutora.

DATOS DEL CONTACTO

Tutora: Ing. Susej Beovides Luis.

Edad: 27 años.

Institución: Universidad de las Ciencias Informáticas (UCI).

Cargo: Jefa de línea del departamento de Realidad Virtual y Visualización.

Título: Ingeniería en Ciencias Informáticas.

Categoría Docente: Asistente.

E-mail: sbeovides@uci.cu

Tutora: Ing. Yadira Ramírez Rodríguez.

Edad: 28 años.

Institución: Universidad de las Ciencias Informáticas (UCI)

Cargo: Jefa de la asignatura de Ingeniería de Software Facultad 5.

Título: Ingeniería en Ciencias Informáticas.

Categoría Docente: Asistente.

E-mail: yramirezr@uci.cu

DEDICATORIA

A:

Mi abuelita, por ser la primera en apoyarme en realizar mis estudios en esta Universidad.

Mis padres, por enseñarme el camino correcto de la vida, por ayudarme a conquistar este su sueño. Por ser los mejores padres del mundo. Gracias a ustedes he llegado hasta aquí. Los quiero.

Mi familia, sin excepción, por estar siempre unida y presente cuando la he necesitado. Por ayudarme y darme fuerzas. En especial a mis tíos, que han sido padres y madres.

David, por estos inigualables 4 años, por toda tu comprensión, por ser especial.

Mis tutoras, por ser para mí, las mejores tutoras del mundo.

Mis amigos, los de verdad, que más que amigos han sido hermanos.



AGRADECIMIENTOS

A:

Mis padres, por brindarme su cariño y comprensión en todo momento.

Mi familia, por apoyarme y confiar en mí.

Mi abuelita, por quererme y consentirme.

David, por enseñarme lo más lindo de la vida.

Mis tutoras, por poder contar con ellas siempre que las necesité, por aguantar mis preocupaciones y nunca decirme un “no”, por ser mis ejemplos como profesionales, por dedicarme horas y subirme el ánimo cuando veía imposibles. Por brindarme su amistad.

Mis amigas Nany, Mado, Amal por permitirme contar con su amistad, por aconsejarme cuando lo necesité. Por ser sencillamente geniales.

Mis amistades, en especial a las chicas de mi apto Aliané, Bislien, Lilian, Lily y Suly, por ser todas para mí una familia jamás las olvidaré. Y a todas las amistades que compartieron conmigo estos 5 años.

Julito, Andy y Ali, mis amigos que tanto quiero, el “team insuperable”, a los tres muchas gracias por compartir conmigo momentos inolvidables. En especial a ti Ali, gracias por ser mi amiga y estar siempre presente.

Mis “papás UCI” Guille y Rubén, por adoptarme como su “hija”; por ser amigos y maestros, por preocuparse y exigirme, por enseñarme mucho de lo poco que se.

Mis profesores, que con sus conocimientos me permitieron llegar hasta aquí.

Mi amigo Ornelo, por compartir conmigo 10 años de estudios, por ser incondicional, siempre ahí en el momento justo. Gracias mi hermanito.

Oswaldo y a Ernesto, por su dedicación y ayuda, hoy entiendo sus exigencias y las agradezco. Gracias por todo.

Yaself y a Brenda, que me brindaron siempre su ayuda incondicional.

Mis compañeros de toda la carrera, por ser geniales y vivir cinco inolvidables años.

A todos los que contribuyeron de una forma u otra ¡Gracias!

RESUMEN

La acelerada demanda del software y la exigencia cada vez mayor de los clientes, ha demostrado la necesidad de desarrollar productos con calidad y formar equipos de proyectos comprometidos y capaces.

Actualmente no se cuenta con un procedimiento ingenieril en el desarrollo de laboratorios virtuales (LV) con fines educativos que establezca los artefactos educativos y pedagógicos tan importantes para este producto, así como las habilidades de cada integrante del equipo de proyecto de acuerdo al rol que desempeña, por ello la presente investigación tiene como objetivo proponer un marco de trabajo ingenieril que rijan el proceso de desarrollo de LV con fines educativos.

Basándose en el marco de trabajo descrito por el ingeniero de software Roger S. Pressman en su libro *Ingeniería de Software: un enfoque práctico* sexta edición del 2005, se detallan cinco fases acopladas al desarrollo de los LV con estos fines, donde se contemplan de igual forma que los aspectos ingenieriles los aspectos educativos. Las mismas se manifiestan a lo largo del ciclo de vida del software y cuentan con acciones de ingeniería y conjuntos de tareas para mantener dichas fases, constituyendo buenas prácticas y criterios para la construcción del software.

Mediante la observación de experiencias pasadas y actuales de desarrollo de laboratorios virtuales con esta finalidad se realiza un análisis comparativo con el desarrollo de un laboratorio virtual base (prototipo no funcional) que usa elementos de esta propuesta con el fin de analizar el comportamiento de uno y otro.

Palabras clave: didáctica, ingeniería de software, laboratorio virtual, marco de trabajo, pedagogía.

ÍNDICE

RESUMEN	III
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABLAS.....	VII
INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	4
1.1 INTRODUCCIÓN	4
1.2 INGENIERÍA DEL SOFTWARE	4
1.3 MARCOS DE TRABAJO PARA EL DESARROLLO DE SOFTWARE	4
1.3.1 SELECCIÓN DEL MARCO DE TRABAJO BASE	11
1.4 TENDENCIAS PARA DESARROLLAR SOFTWARE DE REALIDAD VIRTUAL CON FINES EDUCATIVOS.....	12
1.5 METODOLOGÍAS INGENIERILES PARA SOFTWARE DE RV CON FINES EDUCATIVOS	13
1.6 CONSIDERACIONES PARCIALES.....	14
CAPÍTULO 2. SOLUCIÓN PROPUESTA.	15
2.1 INTRODUCCIÓN	15
2.2 PROYECTO LABORATORIOS VIRTUALES	15
2.2.1 Planeación estratégica	15
2.2.2 Planeación estratégica para el proyecto Laboratorios Virtuales	17
2.3 MARCO DE TRABAJO PROPUESTO.....	20
2.4 FASE DE COMUNICACIÓN.....	20
2.4.1 Ingeniería de Requisitos para laboratorios virtuales con fines educativos.	21
2.4.2 Técnicas de levantamiento de requerimientos.....	25
2.4.3 Elaboración del diseño didáctico	27
2.5 FASE DE PLANEACIÓN	27
2.5.1 Establecer el ámbito del proyecto.....	28
2.5.2 Determinar la factibilidad del producto.....	29
2.5.3 Analizar los riesgos	29
2.5.4 Definir los recursos requeridos	30
2.5.5 Estimación.....	31

2.5.6 Definir estrategias de pruebas.....	33
2.5.7 Desarrollar un plan de proyecto.....	33
2.5.8 Definir mecanismos de seguimiento del programa de trabajo	33
2.6 FASE DE MODELADO	34
2.6.1 Modelado del análisis	34
2.6.2 Modelado del diseño	37
2.7 FASE DE CONSTRUCCIÓN	43
2.7.1 Implementación de código a cada clase a partir del Modelo de Diseño.....	43
2.7.2 Reutilización	44
2.7.3 Pruebas.....	44
2.7.4 Estrategia de prueba para una arquitectura convencional	45
2.8 FASE DE DESPLIEGUE	49
2.8.1 Documentación del producto	49
2.8.2 Entrenamiento a usuarios finales.....	50
2.8.3 Aceptación del Usuario.....	50
2.8.4 Consideraciones parciales.....	50
CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA.	51
3.1 INTRODUCCIÓN	51
3.2 PRESENTACIÓN DEL PROTOTIPO	51
3.2.1 Vistas del prototipo desarrollado.....	51
3.3 ANÁLISIS Y DISCUSIÓN DE LA FASE DE COMUNICACIÓN	53
3.4 ANÁLISIS Y DISCUSIÓN DE LA FASE DE PLANEACIÓN.....	54
3.5 ANÁLISIS Y DISCUSIÓN DE LA FASE DE MODELADO.....	54
3.6 ANÁLISIS Y DISCUSIÓN DE LA FASE DE CONSTRUCCIÓN.....	55
3.7 ANÁLISIS Y DISCUSIÓN DE LA FASE DE DESPLIEGUE.....	55
3.8 VALORACIÓN GENERAL DE LOS RESULTADOS	56
3.9 CONSIDERACIONES PARCIALES	57
CONCLUSIONES GENERALES	58
RECOMENDACIONES	59
BIBLIOGRAFÍA	60
ANEXOS.....	62

ÍNDICE DE FIGURAS

FIGURA 1. MARCO DE TRABAJO DESCRITO POR PRESSMAN.	6
FIGURA 2. MARCO DE TRABAJO PROPUESTO AOSA SPACE.	9
FIGURA 3. SECUENCIA DE LAS FASES DE VT MANAGER Y ORDEN DE EJECUCIÓN.....	11
FIGURA 4. MATRIZ DAFO.....	16
FIGURA 5. TIPOS DE REQUERIMIENTOS.	22
FIGURA 6. PIRÁMIDE DE DISEÑO.	38
FIGURA 7. PATRÓN BASADO EN CAPAS.	39
FIGURA 8. VISTA DEL ÁREA DE PRÁCTICA DEL LABORATORIO.	53

ÍNDICE DE TABLAS

TABLA 1. MATRIZ DAFO	19
TABLA 2. ROLES QUE INTERVIENEN EN EL LEVANTAMIENTO DE REQUISITOS.....	24
TABLA 3. COMPARACIÓN DE LOS RESULTADOS.	56

INTRODUCCIÓN

El desarrollo de software es una actividad reciente en la que existen buenos métodos, pero quizás el problema esté en que no son lo suficientemente difundidos o valorados. El software es actualmente, dentro de cualquier sistema basado en el uso de ordenadores, el componente cuyo desarrollo presenta mayores problemas: es el más difícil de planificar, el que tiene mayor probabilidad de fracaso y el que tiene menos posibilidades de que se cumplan las estimaciones de costes iniciales.

Por otra parte, la demanda de software y su complejidad aumentan continuamente, lo cual maximiza la magnitud de estos problemas. Sólo recientemente, el uso de estas técnicas, métodos y procedimientos están logrando una amplia aceptación y el uso de las mismas ha permitido dar solución a los problemas antes mencionados creando buenas prácticas en el equipo de proyecto y dar un paso de avance a lograr la calidad del software.

El desarrollo de software no es una tarea fácil. La complejidad actual de los sistemas informáticos hace en ocasiones necesario construir software de miles de líneas de código, perfectamente puede hacerlo un grupo de personas, pero la cuestión es qué tan correcto puede hacerlo o qué tanto conocen los programadores de las necesidades de sus clientes, de lo anterior se puede concluir que, no se puede programar sin más, es necesario analizar qué se tiene que hacer, cómo se va a hacer, quiénes lo van a hacer y de qué forma se controla el desarrollo del mismo, con el fin de obtener los resultados esperados.

Cuba no queda exenta del desarrollo de software y para ello cuenta con uno de sus exponentes principales, la Universidad de las Ciencias Informáticas (UCI); la misma está formada por una red de centros especializados en la informatización de las áreas de la sociedad. Entre los mismos se encuentra el Centro de Informática Industrial (CEDIN) de la Facultad 5, en el cual se desarrolla el proyecto Laboratorios Virtuales, que tiene como objetivo el desarrollo de LV que sirvan como apoyo al Proceso de Enseñanza-Aprendizaje (PEA). El proyecto se encuentra desarrollando en estos momentos nueve nuevos productos del mismo carácter y bajo la experiencia alcanzada en el desarrollo se observa:

- Desorden a la hora de realizar el desarrollo del software desde el punto de vista ingenieril y la generación de la documentación.
- El ciclo de vida que sigue el proyecto no contempla que se generen artefactos donde se especifiquen los objetivos y habilidades didácticas con los que debe contar el laboratorio virtual en desarrollo, así como los objetivos y especificaciones técnicas que responden a los objetivos didácticos relevantes en el sistema.

- En el equipo de desarrollo se encuentran recursos humanos (RRHH) realizando actividades que no son las acreditadas a su rol.
- No se tiene bien especificado en ningún artefacto las características de los modelos 3D y 2D que simulan el entorno real.

Debido a estos inconvenientes se hace difícil que se genere la documentación y por lo tanto se pierde tiempo.

Atendiendo a la problemática planteada anteriormente, se define como **problema científico** a resolver: ¿Cómo establecer una organización en el proceso de desarrollo de software de un laboratorio virtual con fin educativo?

Partiendo del problema se define como **objeto de estudio**, marcos de trabajo ingenieriles para el desarrollo de software de realidad virtual.

Se define como **objetivo general** de esta investigación: Elaborar un marco de trabajo ingenieril para el desarrollo de laboratorios virtuales con fines educativos.

Dentro de esta área se definió como **campo de acción** los marcos de trabajo ingenieriles para el desarrollo de LV con fines educativos.

Para lograr el cumplimiento exitoso del objetivo general de esta investigación es necesario realizar las siguientes **tareas investigativas**:

- Levantamiento de información mediante el uso de alguna técnica de recopilación de información para conocer el estado actual de organización del proyecto.
- Elaboración del marco teórico de la investigación respondiendo al estado del arte del tema a investigar para lograr las bases de la investigación.
- Análisis de los procesos de desarrollo de LV con fines educativos.
- Selección del marco de trabajo para el desarrollo de software con los cuales propiciar las bases de la investigación.
- Diseño de un marco de trabajo ingenieril para el desarrollo de LV con fines educativos.
- Confección de los nuevos artefactos a generar, así como sus roles asociados y actividades para organizar la información y declarar nuevas responsabilidades.
- Aprobación de la solución propuesta por parte del equipo de dirección del proyecto.
- Validación parcial de la solución propuesta.

Posibles resultados: Un marco de trabajo ingenieril para el desarrollo laboratorios virtuales con fines educativos que guíe el proceso de desarrollo de software, independientemente de la metodología que se utilice, incluyendo dentro los nuevos artefactos generados, roles y sus responsabilidades.

Para la confección de este trabajo y el cumplimiento total de la siguiente investigación fue necesario hacer uso de los siguientes **métodos científicos de investigación:**

MÉTODOS TEÓRICOS:

- **Analítico – Sintético:** se utiliza para analizar las teorías existentes sobre el estado del arte de los LV así como un estudio detallado del proceso de desarrollo de software, permitiendo la selección de los elementos más importantes que se relacionan con el objeto de estudio.
- **Análisis Histórico – Lógico:** se utiliza para constatar teóricamente cómo ha evolucionado el tema de los LV y el proceso que se sigue para el desarrollo de los mismos.

MÉTODOS EMPÍRICOS:

- **Encuesta:** se utiliza con el objetivo de realizar una encuesta para ver el estado actual del proyecto en cuanto a organización.
- **Entrevista:** se utiliza con el objetivo de realizar entrevistas a los miembros del equipo de proyecto y conocer cómo creen que fuera mejor estructurar el proceso de desarrollo de software.
- **Observación:** se utiliza con el objetivo de realizar una observación en el entorno real del desarrollo de un laboratorio virtual para comparar su comportamiento al ser realizado con el marco de trabajo como guía.

A continuación la estructura de este trabajo incluyendo una breve descripción de cada capítulo:

CAPÍTULO 1 — FUNDAMENTACIÓN TEÓRICA.

En este capítulo se definen los principales conceptos que estarán LV con fines educativos y el proceso de desarrollo de software que siguen los mismos, como también un estudio del arte referente a los marcos de trabajo.

CAPÍTULO 2 — DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

Se propone una solución al problema planteado, haciendo uso de la propuesta de un marco de trabajo ingenieril definiendo actividades aplicables a lo largo del desarrollo como resultado de la investigación. Para dicha propuesta se hace uso del marco de trabajo definido por el profesor Roger S. Pressman.

CAPÍTULO 3 — VALIDACIÓN DE LA PROPUESTA.

Se valida parcialmente la propuesta del marco de trabajo realizando un análisis entre las distintas fases del marco propuesto, tomando el desarrollo de un laboratorio sin utilizar el marco y analizando su comportamiento de acuerdo al desarrollo de un laboratorio base parcialmente guiado por el marco.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se exponen los elementos teóricos que sustentan el objetivo de la investigación. Se exponen elementos generales de la ingeniería de software y dentro de estos, argumentos necesarios para el entendimiento de un marco de trabajo, así como el estudio de algunos marcos propuestos para el desarrollo de software y las tendencias de desarrollo de software de realidad virtual y con fines educativos.

1.2 Ingeniería del Software

Satisfacer las necesidades de clientes exigentes logrando el desarrollo de un software con características tan básicas como la reutilización del mismo y su perfecto funcionamiento, no es un proceso tan sencillo como en ocasiones se suele pensar, como todo proceso, lleva su secuencia de actividades que en este caso pueden ser tan complejas como el equipo de desarrollo pretenda, debido a su conocimiento o no sobre el negocio.

Desarrollar un software y desarrollarlo bien es cada vez un negocio más explotado a nivel mundial, pero no es una tarea de una sola persona, es una elaboración conjunta que requiere de una disciplina, de realizar una ingeniería de software.

El término *Ingeniería de Software*, tiene varias definiciones entre las que se pueden citar:

- *“Ingeniería del Software trata del establecimiento de los principios y métodos de la Ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales.”* (1)
- *“Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software.”* (2)

Para el desarrollo de esta investigación se toma como ingeniería de software la disciplina que aplicando un conjunto de métodos y técnicas a lo largo del desarrollo del software, permite tener como resultado un producto con calidad y responda a las necesidades del cliente.

1.3 Marcos de trabajo para el desarrollo de software

Un marco de trabajo es un conjunto de conceptos, prácticas y criterios para enfocar cualquier problemática; es lo que establece las bases para un proceso de software completo, al identificar un conjunto de actividades que serán aplicables a cualquier proyecto de software independientemente de su tipo, duración o complejidad. Es un conjunto de buenas prácticas

que guía a los integrantes del equipo de trabajo sobre las actividades que tienen que llevar a cabo a lo largo del ciclo de vida del desarrollo de software y que permitan identificar los roles que mejor se ajustan a cada actividad así como las habilidades y conocimientos requeridos de las capacidades tecnológicas necesarias para su implantación y aplicación.

Dentro del proceso del software, se establece un marco común del proceso. En éste se definen un conjunto de tareas aplicables a los proyectos, buscando como finalidad la calidad del software.

Con esta investigación se pretende el uso de un marco que sea específico para el desarrollo de un software de realidad virtual con fin educativo y permitiendo así realizar actividades específicas para ese tipo de producto. En el cual se determinen que técnicas, métodos y documentos se debe de hacer en cada una de las actividades descritas para el marco de acuerdo al objetivo de este.

A continuación se hace un estudio de los marcos identificados en el proceso de investigación que responde al estado del arte del tema, con el objetivo de determinar que marco se toma como base para especificarlo al desarrollo de laboratorios virtuales con fines educativos.

Marco de trabajo de Roger S. Pressman

El ingeniero de software y profesor, Dr. Pressman, en su libro *Ingeniería de Software: un enfoque práctico*, propone un marco de trabajo para formar las bases del desarrollo de un software. Dicho marco de trabajo (*Ver Figura 1*) está compuesto por un conjunto de actividades aplicables a lo largo del proceso del software.

Cada actividad del marco contiene un conjunto de acciones de ingeniería de software, o sea, un conjunto de tareas que producen un producto del trabajo en la ingeniería de software. Cada acción está formada por un conjunto de tareas de trabajo individuales. Las actividades del marco de trabajo genérico para cualquier proceso definidas por Pressman, son las siguientes:

- **Comunicación:** Describe la intensa colaboración y comunicación con los clientes. Incluye la investigación de requisitos así como otras tareas relacionadas.
- **Planeación:** Establece un plan de trabajo de ingeniería de software incluye tareas, técnicas, riesgos probables y recursos necesarios, un programa de trabajo y los productos del mismo.
- **Modelado:** Permite la creación de modelos que permiten al desarrollador y al cliente entender mejor los requisitos del software y el diseño que logrará satisfacerlo.
- **Construcción:** Plantea combinar la generación de códigos (manual o automática) y realización de pruebas necesarias para descubrir errores.

▪ **Despliegue:** Plantea que al entregar el software al cliente este lo evalúa y proporciona información a partir de su evaluación.

Estas cinco actividades genéricas podrán ser utilizadas en el desarrollo de software con diferentes fines, los detalles del software pueden ser muy diferentes, pero estas actividades se mantendrán iguales, para el caso de esta investigación se especifican para el desarrollo de LV con fines educativos.

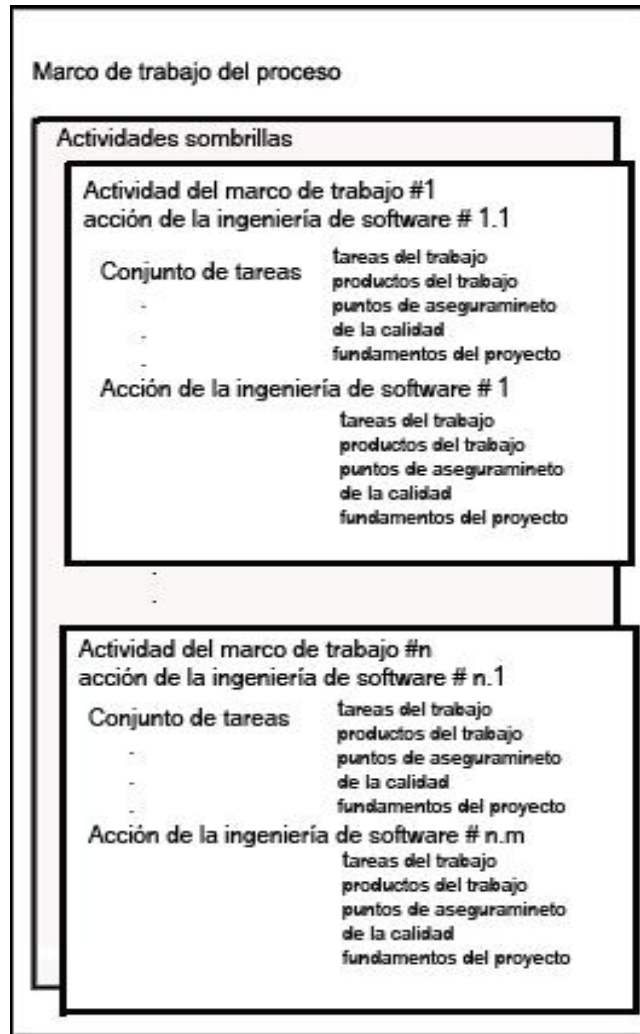


Figura 1. Marco de trabajo descrito por Pressman.

Como anteriormente se explicaba, cada actividad del marco contiene un conjunto de acciones de ingeniería de software. Se observa en la figura 1, que cada acción de la ingeniería de software la representa un gran número de diferentes conjuntos de tareas: una serie de tareas del trabajo, producto del trabajo, puntos de aseguramiento de la calidad y fundamentos del

proyecto. El conjunto de tareas¹ que mejor se ajuste a las necesidades del proyecto es el elegido al final, en conclusión: Cada acción de la ingeniería puede ser adaptable a las necesidades específicas de cada proyecto, así como a las características de su equipo de desarrollo.

Este marco descrito de forma general para el proceso lo completa una serie de *actividades sombrillas*:

- **Seguimiento y control del proyecto de software:** Permite que el equipo de software evalúe el progreso comparándolo con el plan del proyecto y así tomar acciones necesarias para mantener el programa.
- **Gestión del riesgo:** Evalúa los riesgos que pueden afectar los resultados del proyecto o la calidad del producto.
- **Aseguramiento de la calidad del software:** Define y conduce las actividades para asegurar la calidad del software.
- **Revisiones técnicas formales:** Evalúa los productos del trabajo de la ingeniería de software en un esfuerzo encaminado a descubrir y eliminar los errores, evitando que se extiendan hacia la siguiente acción.
- **Medición:** Define y recolecta mediciones del proceso, el proyecto y el producto para ayudar al equipo a entregar software para que satisfaga las necesidades del cliente.
- **Gestión de la configuración del software:** Maneja los efectos del cambio a través del proceso del software.
- **Gestión de la reutilización:** Define los criterios para la reutilización de productos del trabajo y establece mecanismos para la creación de componentes reutilizables.
- **Preparación y producción del producto de trabajo:** Abarca las actividades requeridas para crear productos del trabajo como modelo, documentos, registros, formatos y listas. (3)

El Marco de referencia para la Dirección de Proyectos (PMBOK)

El propósito de esta guía básicamente radica en identificar un conjunto de buenas prácticas que pueden tener un impacto considerable en el éxito de un proyecto, además proporcionar a través del Código de Conducta una guía con las obligaciones básicas de responsabilidad, respeto, imparcialidad y honestidad que deben cumplir los profesionales de esta área. Proporciona y promueve un vocabulario común en el ámbito de la profesión de la dirección de proyectos, para analizar, escribir y aplicar conceptos de la dirección de proyectos. Tener un vocabulario

¹ se define como el conjunto de acciones que deben realizarse para lograr cumplir los objetivos de una acción de la ingeniería de software.

estándar es un elemento esencial en toda disciplina profesional. Se trata de una guía, más que de una metodología. Se pueden usar diferentes metodologías y herramientas para implementar el marco de referencia. El PMBOK reconoce cinco grupos de procesos básicos y nueve áreas de conocimiento comunes a casi todos los proyectos. Los procesos se traslapan e interactúan a través de un proyecto o fase. Son descritos en términos de: Entradas (documentos, planes, diseños y otros), Herramientas y Técnicas (mecanismos aplicados a las entradas) y Salidas (documentos, productos y otros). Las nueve áreas del conocimiento mencionadas en el PMBOK son:

- Gestión de la Integración.
- Gestión del Alcance.
- Gestión del Tiempo.
- Gestión de la Calidad
- Gestión de Costos.
- Gestión del Riesgo.
- Gestión de Recursos Humanos.
- Gestión de la Comunicación.
- Gestión de las Compras y Adquisiciones.

Fortalezas del PMBOK.

- La guía del PMBOK es un marco y un estándar
- Está orientada a procesos
- Indica el conocimiento necesario para manejar el ciclo vital de cualquier proyecto, programa y portafolio a través de sus procesos.
- Define para cada proceso sus insumos, herramientas, técnicas y reportes necesarios (entregables)
- Define un cuerpo de conocimiento en el cual cualquier industria pueda construir las mejores prácticas específicas para su área de aplicación.

Limitaciones del PMBOK.

- Complejo para los proyectos pequeños
- Tiene que ser adaptado a la industria del área de aplicación, el tamaño y el alcance del proyecto, el tiempo y el presupuesto y los apremios de la calidad.

Aunque es un marco de trabajo, directamente está enmarcado en la dirección de proyectos, no resulta aplicable en el contexto que se necesita para el desarrollo de laboratorios virtuales con fines educativos. Por lo tanto se descarta su uso como guía ingenieril en el desarrollo de laboratorios virtuales con fines educativos. (4)

Marco de trabajo para el desarrollo de arquitecturas software orientado a aspectos

La tesis para optar por el doctorado en ingeniería informática presentada por Amparo Navasa Martínez de la Universidad de Extremadura de España, plantea que aunque existen diferentes metodologías para la gestión de equipos de desarrollo software, como el Equipo de Proceso de

Software (TSP, para sus siglas en inglés), se carece de procesos, metodologías o conjunto de buenas prácticas definidas para la gestión de equipos de desarrollo software que trabajan en entornos distribuidos geográficamente, es decir, debe ser una formalización en la que se definan las diferentes actividades que se deben realizar para gestionar de forma eficiente un equipo de desarrollo software global. Esta tesis tiene el objetivo de proporcionar un marco de trabajo para el desarrollo de sistemas software orientado a aspectos (AOSA Space, para sus siglas en inglés), que está formado por un modelo arquitectónico: (AOSA Model, para sus siglas en inglés), una metodología de trabajo, un lenguaje de descripción arquitectónica (LDA) orientado a aspectos y una herramienta para su utilización. La *Figura 2* muestra, a modo de resumen, el marco de trabajo que se presenta como aporte final de esta tesis doctoral. En ella, las elipses con líneas finas representan especificaciones en el lenguaje de modelado (UML); las elipses con líneas gruesas simbolizan la representación formal de la arquitectura (en un LDA). En **negrita y fondo coloreado** se representa el sistema, expresado en UML y su descripción arquitectónica formal. Las flechas representan las acciones ejecutadas por el marco de trabajo. Los pasos que se siguen son los siguientes:

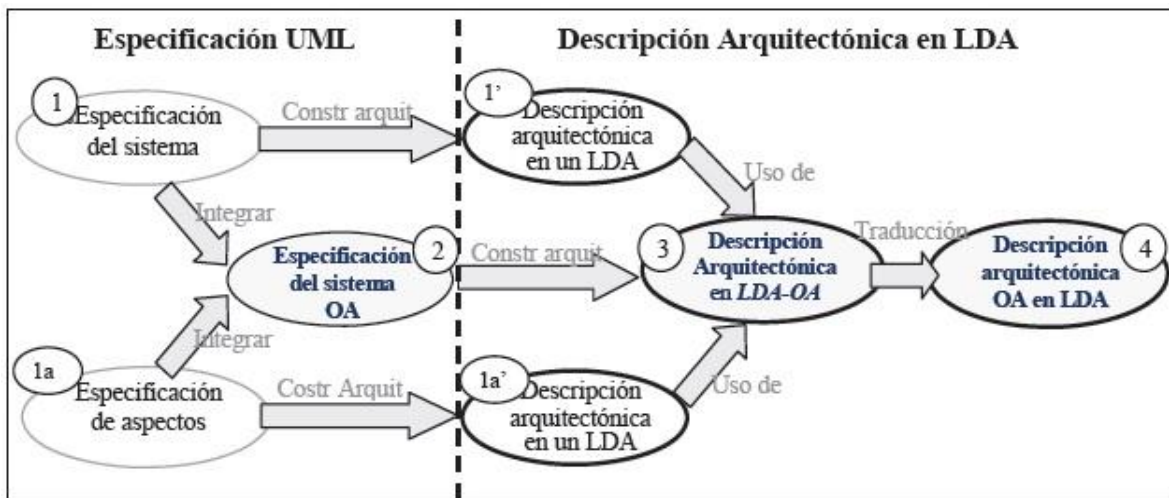


Figura 2. Marco de trabajo propuesto AOSA Space.

- El sistema se representa mediante un conjunto de especificaciones expresadas en su diagrama de casos de uso y los correspondientes diagramas de secuencia. Cada elemento debe describirse completamente a fin de elaborar la oportuna documentación. De este modo se obtiene el punto de vista externo del sistema.
- El diagrama de casos de uso inicial se extiende con los casos de uso que contienen las especificaciones de aspecto (caso de uso extendido). Como consecuencia, los diagramas de secuencia también resultarán modificados. Para realizar la inserción de cada aspecto hay que

considerar cierta información extra: sus condiciones de aplicación, dónde, cuándo y cómo deben aplicarse.

- Realizada la descripción arquitectónica del sistema, se debe expresar en un Lenguaje de Descripción de Arquitecturas Orientado a Aspectos. De este modo, se puede chequear la consistencia y corrección del sistema extendido.
- La representación del sistema en el LDA-OA AspectLEDA se traduce a un LDA convencional (LEDA), que permite a su vez generar un prototipo del sistema cuando se traduce a Java (pues el lenguaje elegido lo permite).(5)

En la descripción de este marco se evidencia que está desarrollado específicamente para las arquitecturas de software orientadas a aspectos (OA), al igual que el PMBOK no es aplicable al desarrollo de los laboratorios virtuales con fines educativos.

Marco para mejorar la gestión de los equipos de desarrollo software global

La tesis doctoral de Javier Saldaña Ramos, define a *VTManager*. Un Marco para la mejora en la gestión de los equipos de desarrollo Software Global. Está dirigido a mejorar la eficiencia en estos equipos, consecuentemente, su rendimiento; mediante la definición e integración de prácticas provenientes de tres disciplinas diferentes, gestión de personas, gestión de proyectos y desarrollo de software. Partiendo de esta visión multidisciplinar, la solución que plantea esta tesis doctoral consiste en elaborar un marco de trabajo que defina el conjunto de buenas prácticas que mejoran la eficiencia de los equipos de desarrollo de software que se encuentran dispersos geográficamente e incrementar el rendimiento de los mismos, de forma que sea fácil y aplicable y que permita, por tanto, a las organizaciones afrontar con mayor probabilidad de éxito proyectos con múltiples participantes dispersos geográficamente, asegurando la calidad del producto final y minimizando los costes. Este marco de trabajo contiene también la definición de los principales roles y responsabilidades que existen en un equipo de desarrollo software global así como un modelo de competencias con las habilidades, capacidades y conocimientos requeridos para cada uno de ellos. El objetivo principal del marco es proporcionar un conjunto de prácticas eficientes para formar, desplegar y gestionar equipos de desarrollo de software global, desde las primeras etapas de construcción del equipo hasta su disolución y la reintegración de sus miembros en otros equipos de trabajo.

El modelo de ciclo de vida propuesto para la gestión del equipo de desarrollo de software global consta de cinco fases principales (*Ver Figura 3*) donde para cada una de ellas se han definido el conjunto de buenas prácticas, así como la secuencia en la que éstas deben ser ejecutadas, teniendo en cuenta los roles que intervienen y las capacidades tecnológicas adecuadas. De modo que, independientemente de la metodología de desarrollo y de gestión de proyectos

aplicadas, se disponga de un conjunto de buenas prácticas; estableciendo, para cada una de las prácticas definidas, las capacidades que deberían implementar las tecnologías y herramientas de soporte al trabajo del equipo y las competencias que deberían reunir los integrantes del equipo de trabajo que participen, de acuerdo al rol que desempeñan en la misma. (6)

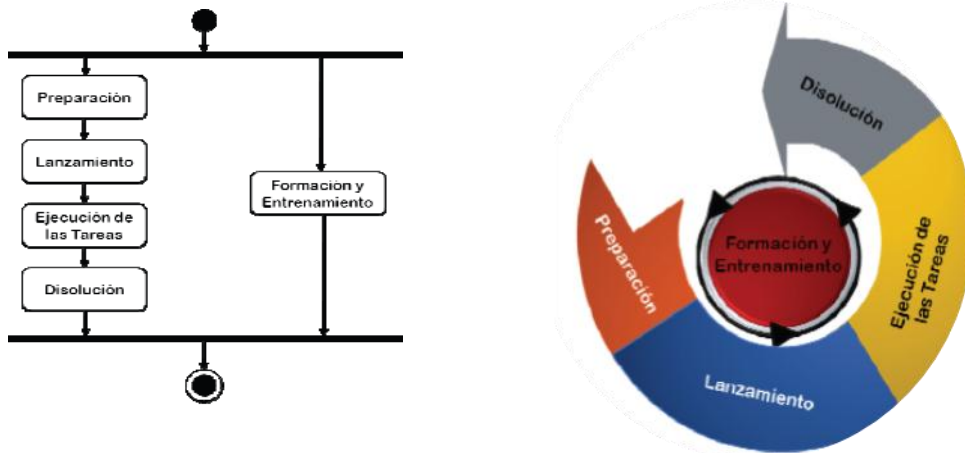


Figura 3. Secuencia de las Fases de VT Manager y orden de ejecución.

Este último define lo que en verdad necesita el equipo de proyecto para realizar el producto, pero no resulta candidato porque está detallado para un área en específico, la gestión de los equipos de desarrollo software global, que no aplica para el desarrollo que tiene esta investigación como propósito, laboratorios virtuales con fines educativos.

1.3.1 Selección del marco de trabajo base

Luego de haber analizado los marcos de trabajo antes expuesto e identificar su utilidad, se evidencia que la mayoría han sido especificados para temas que no comporten similitudes con LV con fines educativos. Únicamente el marco de trabajo de Pressman plantea sus actividades genéricas, flexibles e independientes de cualquier producto, que permiten ser aplicadas a cualquier proceso de desarrollo de software, además propone un conjunto de tareas para cada una de esas actividades.

Por lo tanto para esta investigación se toma como base el Marco de Trabajo de Pressman, para ser detalladas cada una de estas actividades al desarrollo de LV con fines educativos.

1.4 Tendencias para desarrollar software de Realidad Virtual con fines educativos.

La Realidad Virtual (RV) es una tecnología adecuada para la enseñanza, debido a su facilidad para captar la atención de los participantes mediante su inmersión en mundos virtuales relacionados con las diferentes ramas del saber, lo cual puede ayudar en el aprendizaje de los contenidos de cualquier materia.

En el estudio del arte realizado tanto en su búsqueda nacional como internacional, **no se ha encontrado definido ningún marco de trabajo** que sea específico para RV que se pueda referenciar y tomar como base para esta solución.

Por lo tanto las tendencias para el desarrollo de este tipo de producto se reducen a la utilización de los mismos estilos que se usan para el desarrollo de software de otro corte, entre las que se destacan:

- Las metodologías de desarrollo de software.
- La automatización de los métodos de la ingeniería de software.
- La programación Orientada a Objeto (POO). (7)

Aunque no es objetivo tratar el tema de las metodologías pues más bien el hilo de la investigación está centrado en los marcos de trabajo, sí se analiza cómo se guía su desarrollo aunque en ocasiones no sea el más recomendado, por lo que se hace importante tratar que actualmente también el paradigma seguido para el desarrollo de este tipo de software es el uso de metodologías de desarrollo de software, ya sean ágiles como: SCRUM², o la más difundida Programación Extrema (XP, para sus siglas en inglés) todas orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando.

Similar a lo anterior y en dependencia de las características del software se usan metodologías tradicionales, las mismas se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida, uno de sus exponentes principales es Proceso Unificado de Software (RUP, para sus siglas en inglés) sus características fundamentales son: guiado por casos de uso, centrado en la arquitectura, iterativo e incremental.

² La palabra SCRUM no son siglas, sino que significa *melé*, un tipo de jugada del rugby.

1.5 Metodologías ingenieriles para software de RV con fines educativos

El software educativo se ha convertido en una necesidad para el desarrollo del PEA tanto para el sistema educacional cubano como para el internacional. Definiéndose como:

▪ *Un software educativo es un programa para ordenador creado con el fin de ser utilizado como medio didáctico, que pretende imitar la labor tutorial que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los alumnos. Por lo tanto, está centrado en el Proceso de Enseñanza- Aprendizaje y pretende atender las necesidades del estudiantado en función de los programas educativos.*(8)

Atendiendo a la definición antes expuesta y llegando a una conclusión el software educativo es aquel que solamente o en conjunto con otros recursos audiovisuales, es capaz de apoyar y guiar el aprendizaje del usuario que con el interactúa. Se caracteriza por ser interactivos, individualizar el trabajo de los estudiantes y ser fácil de usar. Dentro del software educativo se encuentran los laboratorios virtuales, siendo los medios educativos asistidos por computadoras con más auge a nivel mundial, los mismos son sistemas no inmersos de realidad virtual y son definidos como:

Un espacio electrónico de trabajo concebido para la colaboración y la experimentación a distancia con objeto de investigar o realizar otras actividades creativas y elaborar y difundir resultados mediante tecnologías difundidas de información y comunicación (9).

Para el desarrollo de software educativo se han hecho numerosos aportes en cuanto a metodologías, por las características especiales que este tipo de software presente. A continuación una síntesis de las más referenciadas:

▪ **Metodología de ingeniería de software educativo por Álvaro Galvis Panqueva:** Galvis Panqueva en 1991 desarrolló una metodología para el desarrollo de software educativo que consta de 5 etapas: análisis, diseño, desarrollo, prueba piloto y prueba de campo. Esta metodología puede tener dos maneras de ejecución, en función de los resultados de la etapa de análisis: en el sentido de las manecillas del reloj se procede a diseñar, desarrollar y probar lo que se requiere para atender una necesidad. En el sentido contrario, se someta a prueba aquello que se encontró y puede satisfacer la necesidad.

▪ **Metodología para la elaboración de software educativo definida por Pere Marqués:** La metodología para la elaboración de software educativo definida por Marqués en 1995 contempla 11 etapas de desarrollo, cada una de las cuales se puede dividir en fases más específicas. Estas etapas principales son:

- Génesis de la idea.
- Pre-diseño o diseño funcional.

- Estudio de la viabilidad y marco del proyecto.
- Dossier completo del diseño o diseño orgánico.
- Programación y elaboración del prototipo
- Redacción de la documentación del programa.
- Evaluación interna.
- Ajuste y elaboración del prototipo.
- Evaluación externa.
- Ajuste y elaboración de la versión.
- Publicación y mantenimiento del producto.

Estas metodologías representan las tendencias en el desarrollo de soluciones informáticas educativas, las mismas se inclinan hacia los aspectos educativos que tiene el software, pero no son explícitos en las actividades de la ingeniería del software, muestran sus actividades de forma general sin caer en los elementos exactos de ingeniería tan importantes para el desarrollo, como es el caso de no proponer realizar un análisis o un diseño, se van directamente a la implementación del software.

Otro elemento a tener en cuenta es que las mismas fueron concebidas hace algunos años y el desarrollo de software y sus métodos es acelerado donde ya hoy no son factibles pues la demanda requiere otros métodos.

Además de estas razones se aclara que no se pretende sustituir una metodología u otra sino establecer una organización en la estructura del equipo de proyecto que permita guiar el proceso de desarrollo de laboratorios virtuales con fines educativos.

1.6 Consideraciones parciales.

En el desarrollo de este capítulo se llegaron a consideraciones como:

- Se establecieron las bases necesarias para la comprensión del objeto de estudio y su campo de acción.
- Se analizaron los conceptos fundamentales concernientes a los marcos de trabajos ingenieriles, la ingeniería de software en general y los laboratorios virtuales como software educativos de RV.
- Se toma como base para la propuesta de solución el marco descrito por Pressman luego del estudio realizado.

CAPÍTULO 2. SOLUCIÓN PROPUESTA.

2.1 Introducción

El presente capítulo parte de una planeación estratégica para determinar el estado actual del proyecto. Se describen las fases del marco de trabajo para el desarrollo de LV con fines educativos basándose en el descrito por Pressman. Se detalla para cada fase las actividades que se deben realizar y dentro de las mismas los roles encargados y artefactos a generar.

2.2 Proyecto Laboratorios Virtuales

El marco de trabajo debe ser extensible para cualquier entidad que se dedique al desarrollo de este tipo de producto, pero es válido aclarar que como escenario se toma el proyecto Laboratorios Virtuales, el mismo es un proyecto productivo del Centro de Informática Industrial que se encuentra actualmente en el cumplimiento de su objetivo principal: Desarrollar laboratorios virtuales para la educación. Contar con estos laboratorios virtuales, permitirá fortalecer las habilidades intelectuales y prácticas de poblaciones masivas de estudiantes en el desarrollo de procesos simulados. En el análisis del mismo se han encontrado los problemas reflejados en la situación problémica, que perfectamente puede ser el caso de cualquier otro proyecto con el mismo objetivo. El equipo de proyecto está compuesto por:

- Un asesor de la calidad.
- Un planificador.
- Un arquitecto de software.
- Un Administrador de la configuración.
- Jefe de proyecto.
- Administradores de la calidad.
- Analistas.
- Diseñadores gráficos.
- Programadores.

Para conocer el estado actual referente a la organización de la documentación, las relaciones interpersonales y las capacidades de desempeño en los roles en el que se encuentra dicho proyecto que se toma como muestra, se aplica una encuesta con el objetivo de reconocer las debilidades, fortalezas, amenazas y oportunidades que el equipo de proyecto considera para corroborar la necesidad del marco de trabajo en el desarrollo de laboratorios virtuales con fines educativos.

2.2.1 Planeación estratégica

“La planeación estratégica es un conjunto de acciones que deben ser desarrolladas para lograr los objetivos estratégicos, lo que implica definir y priorizar los problemas a resolver, plantear

soluciones, determinar los responsables para realizarlos, asignar recursos para llevarlos a cabo y establecer la forma y periodicidad para medir los avances” (10). Tiene como objetivo trazar un mapa de la organización, que señale los pasos para alcanzar la visión y convertir los planes en acciones. Define elementos como:

Misión: Expresa la razón de ser de su empresa o área; es la definición del negocio en todas sus dimensiones. Involucra al cliente como parte fundamental del negocio.

Visión: Es un conjunto de ideas generales que determinan el marco de lo que una empresa es o quiere ser en el futuro.

Diagnóstico estratégico: El diagnóstico estratégico revela las particularidades específicas del sistema en cuestión en el momento en que se realiza el ejercicio de proyección estratégica. Se realiza empleando la técnica de trabajo en grupo, Matriz DAFO (Matriz de Debilidades, Amenazas, Fortalezas y Oportunidades).



Figura 4. Matriz DAFO.

2.2.2 Planeación estratégica para el proyecto Laboratorios Virtuales

El diseño estratégico se realiza con el fin de afirmar la organización, descubrir lo mejor de ella y aclarar ideas futuras. La planeación estratégica cuenta con un conjunto de elementos que se describen a continuación. Luego de haber aplicado la encuesta se determinó lo siguiente.

- **Misión:** permitir que las personas y las entidades que soliciten los servicios que se brindan aprecien su potencial. Producir soluciones informáticas (laboratorios virtuales con fines educativos) que permitan apoyar el PEA y elevar el prestigio de la industria cubana del software
- **Visión:** desarrollar laboratorios virtuales en cualquier área de estudio, con calidad y que satisfagan las necesidades de clientes exigentes.

Debilidades (D):

- D1 Exceso de reuniones en el equipo de proyecto.
- D2 Problemas en la captura de requisitos.
- D3 No hay suficiente conocimiento de las materias de las cuales se desarrollan los LV.
- D4 Desconocimiento del trabajo con la tecnología.
- D5 El proceso de desarrollo del software no cumple las expectativas al no considerar aspectos educativos.
- D6 Se realiza la integración de los módulos por integradores con falta de capacitación.
- D7 Cambios constantes en la planificación del desarrollo del producto.
- D8 No se respetan las fases de desarrollo.
- D9 Los recursos humanos no se encuentran en la misma área de trabajo.
- D10 Se realizan pocas revisiones y chequeos al proyecto.
- D11 Hay poca comunicación con el cliente por la distancia en el país o fuera de él.
- D12 El equipo de proyecto no está a tiempo completo en la producción del software.
- D13 Falta de recursos materiales.

Amenazas (A):

- A1 Competencia por parte de otras entidades que se dedican al desarrollo de sistemas de este tipo.
- A2 Problemas de presupuesto destinado al proyecto.
- A3 Fracaso del producto por desorganización en el proceso de desarrollo de software.

Fortalezas (F):

- F1 Se cuenta con suficiente experiencia en el desarrollo de laboratorios virtuales.
- F2 Existe comunicación entre sus integrantes de proyecto.
- F3 Alto grado de compromiso de los miembros del proyecto con el trabajo.

- F4 Existe entre el equipo de proyecto buena práctica del trabajo.
- F5 Se requieren procesos de desarrollo de software diferentes para cada laboratorio.
- F6 Equipo con altas expectativas de superación e investigación.

Oportunidades (O).

- O1 Posibilidades de contratos de desarrollos con otras entidades para la producción de laboratorios virtuales.

Combinando cada una de estas variables se puede realizar la matriz DAFO.

Matriz DAFO		Oportunidades	Amenazas			Total
		O1	A1	A2	A3	
Fortalezas	F1	X	X	X	X	4X
	F2					
	F3	X				1X
	F4					
	F5	X		X		2X
	F6	X				
Debilidades	D1					
	D2					
	D3	X	X			2X
	D4	X	X			
	D5	X	X		X	3x
	D6		X			1X
	D7				X	1X
	D8				X	1X
	D9				X	1X
	D10					
	D11				X	1X
	D12	X	X		X	3X

	D13		X			
Total		8X	5X	2X	7X	

Tabla 1. Matriz DAFO

▪ **Problema estratégico general:** De continuar con un proceso de desarrollo que no cumple las expectativas en los aspectos educativos y didácticos y un equipo de desarrollo que no está a tiempo completo en la producción del software porque también se dedica a asuntos docentes, aunque cuente con experiencia en el desarrollo de laboratorios virtuales y no se requieran procesos de desarrollo diferentes para cada laboratorio, no le permitirá al equipo la posibilidad de realizar contratos con otras entidades para el desarrollo de laboratorios virtuales con fines educativos.

▪ **Solución estratégica general:** Si se utiliza plenamente la experiencia que tiene el equipo de proyecto en el desarrollo de laboratorios virtuales teniendo en cuenta además, que para todos se realiza el mismo proceso de desarrollo, se podrá aprovechar para realizar contratos con otras entidades y se minimizará el efecto de la competencia de otras empresas que se dediquen al mismo fin, así como evitar el fracaso del mismo y superarse en cuanto al proceso de desarrollo que actualmente, no cumple las expectativas educativas y didácticas.

Atendiendo al problema estratégico planteado anteriormente se puede realizar un análisis de la situación actual en cuanto a la organización en el proceso de desarrollo de los laboratorios virtuales y se evidencia que no está estructurado como realmente se necesita, la estructura ingenieril que se lleva a cabo actualmente no ha podido suplir a plenitud las necesidades de un software de este tipo.

Una de las debilidades más fuertes encontradas es que la metodología que se usa no cumple con algunos aspectos necesarios y siendo así, se plantea en este capítulo la propuesta de un marco de trabajo que se centralice en el desarrollo de laboratorios virtuales con fines educativos. El marco será una especificación del marco de trabajo descrito por Pressman, pero atendiendo exclusivamente a las cinco actividades generales, no se adentra en las especificaciones de las actividades sombrillas, porque las mismas ocurren a lo largo del proceso de software y se enfocan de modo principal en la gestión, el rastreo y el control de proyecto, en sentido general se enfocan en la gestión de proyectos y no en los elementos ingenieriles.

2.3 Marco de trabajo propuesto

El marco ha de dotar al equipo de desarrollo de las habilidades y los conocimientos necesarios para desarrollar laboratorios virtuales con fines educativos. Cuenta con cinco fases: **Comunicación, Planeación, Modelación, Construcción y Despliegue**, dichas fases son una especificación del marco propuesto por el ingeniero de software Roger S. Pressman que permiten organizar el proceso de desarrollo estableciendo la documentación de corte pedagógico que se necesita, así como los roles encargados para documentar la misma. Se estudian además, las técnicas de estimación de proyectos y las técnicas para el levantamiento de requisitos, se establecen aspectos para evaluar los laboratorio virtuales y otros elementos específicos para cada actividad.

2.4 Fase de Comunicación

El proceso de desarrollo de software requiere de una comunicación clara, precisa y muy eficiente. Cada uno de los integrantes del equipo de desarrollo del software debe saber claramente cuál es su función, qué va a encontrar como materia prima para su fase y qué espera la siguiente fase como resultado de su trabajo.

En las relaciones cliente, proveedor y usuarios en los proyectos de desarrollo de software en muchos casos la acción va muy por delante de la comunicación. Se toman decisiones de manera unilateral, no se homogenizan percepciones, no se dialoga, no se informa y lo que es peor, no se escucha. Si se piensa que siempre se lleva la razón, que se tiene todo claro, se está perdiendo toda la capacidad de mejorar lo que proporcionan los demás. Es necesario que exista una comunicación entre las partes involucradas.

Pressman en su marco de trabajo describe esta fase como la intensa colaboración con los clientes y que además debe de incluir la investigación de requisitos así como otras tareas relacionadas, pero debe de quedar bien claro que la comunicación no debe de estar presente solamente en las fases iniciales del desarrollo, debe de realizarse a lo largo de todo el proceso y no únicamente en el levantamiento de requisitos de software, aunque esta sea de vital importancia. La fase de comunicación está compuesta por la acción de ingeniería: Levantamiento de requisitos.

2.4.1 Ingeniería de Requisitos para laboratorios virtuales con fines educativos.

“...la parte más difícil de construir un sistema de software es decidir qué construir, ninguna otra tarea perjudica tanto al sistema, cuando se hace mal; ninguna otra tarea es tan difícil de rectificar posteriormente”. (11)

Dentro de la fase de Comunicación la acción de ingeniería de mayor importancia en el marco de trabajo que se propone es la captura de requisitos; un software en su desarrollo cuenta con requisitos, que para el mejor entendimiento del analista de sistemas, tienen una clasificación que permite conocer las características que posee el requerimiento dentro de la organización. La clasificación para los requerimientos de software se muestra en la figura 5 (*Ver figura 5*).

Al centrarse en el tipo de software que se analiza, es evidente que si se realiza un levantamiento de requisitos como a otro software común, no se analizarían los aspectos educativos y didácticos, lo que no quiere decir que no se tomen los antes mencionados, sino que a estos se les añadan un levantamiento de las características pedagógicas con que debe contar el sistema en cuestión.

Lo primero a tener en cuenta es el levantamiento de los requerimientos, que por ser una actividad completamente humana, debe ser realizado por una persona de gran capacidad comunicativa que le permita poder definir lo que realmente el cliente necesita, permitiendo así establecer el alcance real del producto a realizar.

En el caso de los Laboratorios Virtuales esta tarea es desarrollada por el analista de sistemas, pero atendiendo a que los software que desarrolla el proyecto Laboratorios Virtuales tienen fines educativos, el analista de sistema debe contar con los conocimientos generales de la ingeniería de software en su sentido amplio, pero no cuenta con los factores educativos a los que debe de estar enfocado el producto.

Atendiendo a lo anterior se hace necesario que en la captura de requisitos participe además del analista del sistema una persona con conocimientos pedagógicos que recoja estos elementos con los que también debe contar el software.

La obtención de aspectos pedagógicos en este tipo de programas, es una de las tareas principales en el desarrollo de software con fines educativos, su calidad didáctica va definida por la coherencia entre los objetivos a alcanzar, contenidos a tratar, actividades desarrolladas por los alumnos y las propuestas por el programa.

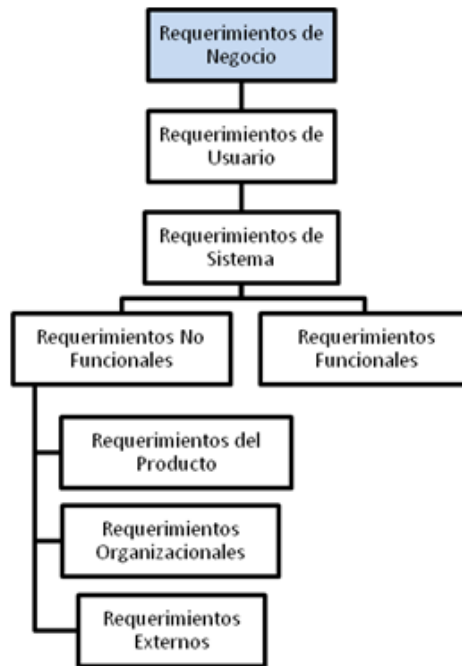


Figura 5. Tipos de Requerimientos.

Por lo antes expuesto se propone que esta acción de la ingeniería de software (levantamiento de requisitos) la realice el analista del sistema y un metodólogo, que en su conjunto realicen la captura desde el punto de vista funcional, no funcional y pedagógico. Luego de realizar consultas a diferentes fuentes se han establecido un conjunto de habilidades y responsabilidades en los aspectos educativo del software. (Ver tabla 2)

Rol	Habilidades.	Responsabilidades.
Analista de sistema.	<ul style="list-style-type: none"> - Habilidades de comunicación -Capacidad de redacción y concreción. -Habilidades de trabajo en equipo. -Conocimiento de metodologías de desarrollo de software. -Conocimiento de ingeniería de software. 	<ul style="list-style-type: none"> -Participa con el cliente y el usuario final recogiendo las entradas de los involucrados relevantes. -Captura los requisitos. -Define las prioridades. -Realiza la especificación de requisitos. -Realiza el seguimiento de los requisitos.

		<p>Participa en el diseño de la solución.</p> <ul style="list-style-type: none"> - Diseña las pruebas. -Participa en la elaboración del Plan de Administración de Requisitos -Determina los proveedores válidos de requisitos - Crea y actualiza Matriz de Trazabilidad(12)
<p>Metodólogo.</p>	<ul style="list-style-type: none"> - Habilidades de comunicación -Capacidad de redacción y concreción. -Habilidades de trabajo en equipo. -Conocimientos y experiencias didácticas. -Conocer el proceso docente educativo, el trabajo metodológico, pedagógico, psicológico, político ideológico y moral bajo la óptica de lo creativo vivencial en la formación integral de la personalidad de los educandos (13) 	<ul style="list-style-type: none"> -Identifica la necesidad del programa educativo. -Define los objetivos educativos. -Define el contenido educativo. -Valora las características de los destinatarios. -Define y revisa en profundidad la estrategia pedagógica y metodológica que se sigue en el software educativo. - Confecciona el método de evaluación. -Planifica la documentación didáctica. -Gradúa el nivel de complejidad de cada uno de los ejercicios así como varía la tipología de las actividades. -Proporciona instrumentos

		<p>de análisis y de diseño pedagógicos.</p> <p>-Elabora el diseño didáctico.</p> <p>-Valida los recursos audio visuales.</p> <p>(13)</p>
--	--	--

Tabla 2. Roles que intervienen en el levantamiento de requisitos.

Es importante de igual manera analizar que en este tipo de sistema es más que necesaria una buena representación visual de las interfaces. Los LV con fines educativos, como cualquier otro software con el mismo propósito, se caracterizan por ser completamente interactivos con los usuarios a los cuales está destinado.

El diseño del producto debe ser lo más explícito posible, cada laboratorio cuenta con una serie de elementos (generalmente herramientas de trabajo) que a la hora de levantar sus requisitos se hace un poco engorroso el proceso, pues se torna difícil ser tan claro en cuanto a las características (tamaño, color) de los mismos, para ello aunque no es del todo necesario, pero sí aconsejable, debe participar un diseñador gráfico y ser capaz de tomar cuáles son las solicitudes para el diseño de cada uno de los elementos que se encontrarán en el producto y hacer el proceso de modelado de las escenas más sencillo. Para tomar estas características el diseñador puede valerse de la toma de fotografías del mundo real que se quiere simular en el entorno virtual.

Todos los datos que se obtienen para cada uno de los elementos que en su conjunto componen el laboratorio deben ser almacenados para lograr mayor organización, por lo tanto se propone una planilla bajo el nombre de Solicitud para diseño (*Ver Anexo 1*), con el objetivo siguiente: describir a detalles cada uno de los elementos que se manejan en el laboratorio virtual; dichos elementos estarán organizados por actividades o por ejercicios según como el diseñador comprenda. Esta planilla debe de ser generada al mismo tiempo que la especificación de requisitos.

La misma posibilita al diseñador una mejor comprensión de lo que se modelará como entorno virtual, además de permitir con la información visual que se recoge el cliente tenga al menos una mínima idea de los elementos que tendrá en su producto final y en caso que no esté de acuerdo, se encuentra en el momento preciso para aclarar que realmente necesita.

2.4.2 Técnicas de levantamiento de requerimientos

Existen varias técnicas para realizar el levantamiento de los requisitos, que permiten lograr una mejor comprensión de las necesidades del cliente y facilitan al ingeniero de software la captura de las mismas. Aunque se explican muchas técnicas, no significa que deban usarse todas, se eligen aquellas que respondan al propósito del sistema. Si dos técnicas sirven para el mismo objetivo, se escoge la de menor costo. Si se quiere estar seguro de cubrir todo, pueden usarse varias técnicas. Lo que pueda faltar en una, lo complementará la otra.

▪ **Las entrevistas:** Son la técnica de elicitación más utilizada y de hecho son prácticamente inevitables en cualquier desarrollo ya que es una de las formas de comunicación más natural entre personas. La entrevista es buena para obtener conocimientos acerca del trabajo actual en el dominio dado y los problemas presentes.

Entrevistas no estructuradas: Permite que el entrevistador formule preguntas no previstas durante la conversación. El entrevistador inquiriere sobre diferentes temas a medida que se presentan. En este tipo de entrevistas el entrevistador tiene poco control sobre el proceso y las conversaciones son enfocadas hacia un tema particular.

Entrevistas estructuradas: Se basan en un marco de preguntas predeterminadas. Las preguntas se establecen antes de que inicie la entrevista y todo solicitante debe responderla.

Este enfoque mejora la calidad de la entrevista, pero no permite que el entrevistador explore las respuestas interesantes o poco comunes. Por eso la impresión del entrevistado y entrevistador es la de estar sometidos a un proceso mecánico. Es posible incluso que muchos solicitantes se sientan desalentados al participar en este tipo de proceso.

Entrevistas semiestructuradas: Es una estrategia mixta, con preguntas estructurales y con preguntas no estructurales. La parte estructurada proporciona una base informativa. La parte no estructurada añade interés al proceso y permite un conocimiento inicial de características específicas.

▪ **Desarrollo Conjunto de Aplicaciones (JAD, para sus siglas en inglés):** Es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo de uno o varios días. En estas reuniones se ayuda a los usuarios y otros beneficiarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.(14)

Esta técnica aunque es muy usada y tiene como beneficio que implica más a los usuarios y otros beneficiarios en el desarrollo pero es válido tener en cuenta que tiene como desventaja

que es muy costosa comparado con otras técnicas tradicionales y que además puede ser incómodo si el grupo es demasiado grande concerniente al tamaño del proyecto y que no siempre se puede contar con la presencia de los usuarios.

▪ **Tormenta de ideas:** Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Las sesiones suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el facilitador de la sesión, encargado más de comenzar la sesión que de controlarla. Como técnica de elicitación de requerimientos, puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requerimientos son todavía muy difusos.

Esta técnica aunque es muy fácil de aprender y requiere poca organización lleva la desventaja si en las sesiones no surge la creatividad, por otro lado, al ser un proceso poco estructurado, puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

▪ **Cuestionarios:** documento formado por un conjunto de preguntas que deben estar redactadas de forma coherente y organizada, secuenciada y estructurada de acuerdo con una determinada planificación, con el fin de que sus respuestas puedan ofrecer toda la información que se precisa.

Los resultados del cuestionario pueden ser un poco difíciles de interpretar. Las personas consultadas pueden haber mal interpretado las preguntas y de igual forma se puede mal interpretar las respuestas. Durante la entrevista, se puede comprobar si se está entendiendo lo planteado y reformular preguntas. También es difícil clasificar los resultados de las preguntas abiertas si no se conoce el contexto.

Un problema con los talleres es que los participantes pueden comprometerse mucho con la solución allí planteada, porque en su entusiasmo, se pueden olvidar otros aspectos, tales como si la solución es técnicamente factible, o si otros actores entienden la solución.

Luego de haber analizado cada una de las técnicas antes expuestas, sus ventajas y desventajas se procede a proponer que en lo que a esta investigación concierne se toma como técnica de levantamiento de requisitos la entrevista y para ir más a lo específico la entrevista semiestructurada, pues resulta apropiada para establecer preguntas concretas sobre un tema determinado o dejar surgir las mismas en el encuentro, se le realiza al personal de interés, no resulta costosa su puesta en práctica, permite establecer un lenguaje común entre las partes implicadas, es fácil de realizar su análisis a diferencia de los cuestionarios que pueden resultar difíciles de interpretar. Aunque la técnica JAD resulte tentadora tiene orientación hacia los

usuario, cuando no siempre es posible contar con ellos en determinado momento, en caso de ser posible puede ser utilizada también, al final es una variación de la entrevista.

Una vez obtenidos los requerimientos, es pertinente organizarlos y representarlos para poder determinar el grado de importancia de cada uno de ellos, de acuerdo a esta importancia se puede determinar la secuencia para ejecutar cada uno y el valor que tiene para el cliente teniendo en cuenta el costo y riesgo de implementación ante los otros requerimientos.

2.4.3 Elaboración del diseño didáctico

Como se abordó en temas anteriores el metodólogo se encarga de obtener las características pedagógicas del software, para ello es responsable de generar el artefacto que a continuación se describe.

El diseño didáctico es el primer guión del software que contiene los aspectos pedagógicos como el contenido, los objetivos generales y didácticos así como la estrategia didáctica a seguir (*Ver Anexo 2*). Este pre-diseño posibilita conocer una primera idea general de lo que se propone con el software que se desarrollará. La técnica entrevista será la utilizada para realizar este documento. Contará con los siguientes elementos y divididos en dos espacios, los aspectos didácticos y los técnicos.

Finalizada la etapa del pre-diseño del programa, el equipo de proyecto en específico el metodólogo siendo este el rol encargado de realizar esta actividad y generar el artefacto descrito con anterioridad, hace llegar el documento al jefe de proyecto que, con el adecuado asesoramiento técnico, pedagógico y comercial, analiza su viabilidad y en caso favorable, establecerá el marco para el desarrollo del proyecto (presupuesto, planificación y otros).

2.5 Fase de Planeación

Luego de haber concluido con la fase de comunicación y todos los elementos que ella propone para el producto que se desea desarrollar, el próximo paso es planificar los pasos que se van a seguir. La planificación es fundamental en el proceso de desarrollo de un producto de software (y en cualquier otro). En el mismo se establece, entre otros, qué tareas y cuándo se van a realizar y los recursos que utilizarán las mismas. En base al plan de trabajo la gerencia podrá determinar si el proceso está marchando en tiempo y si está utilizando los recursos (tiempo, personal, dinero y otros) de la forma esperada. Pressman define la fase de planeación como la actividad que establece un plan para el trabajo de la ingeniería del software y que describe las tareas técnicas que deben de realizarse, los riesgos probables, los recursos que serán requeridos, los recursos de trabajo que serán producidos y un programa de trabajo.

La planificación abarca cinco grandes actividades: la estimación, el programa de trabajo, el análisis de riesgos, la planificación de la gestión de la calidad y la planificación de la gestión del cambio, que se resumen en un conjunto de tareas:

- Establecer el ámbito del proyecto.
- Determinar la factibilidad.
- Analizar los riesgos.
- Definir los recursos requeridos.
 - Determinar los recursos humanos requeridos.
 - Definir lo recursos de software reutilizables.
 - Identificar los recursos del entorno.
- Estimar costo y esfuerzo.
 - Descomponer el problema.
 - Desarrollar dos o más estimaciones empleando tamaño, puntos de función, tareas de proceso o casos de uso.
 - Reconciliar las estimaciones.
- Desarrollar un plan de proyecto.
 - Establecer un conjunto de tareas significativo.
 - Definir una red de tareas.
 - Usar herramientas de planificación para desarrollar un cronograma.
 - Definir mecanismos de seguimiento del programa de trabajo.

En las secciones siguientes se analizará cada una de estas actividades enunciadas por Pressman.

2.5.1 Establecer el ámbito del proyecto

Es la primera actividad llevada a cabo en la planeación del software el ámbito del software describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad, se evalúan las funciones del ámbito y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación, define el contenido que representa para los usuarios el uso del software, el ámbito se define al usar una de las técnicas siguientes:

- Después de una comunicación con todos los participantes se desarrolla una descripción narrativa del ámbito de software.
- Los usuarios finales desarrollan una serie de casos de usos.

En el escenario en cuestión (Laboratorios Virtuales), puede ser aplicada perfectamente cualquiera de las dos técnicas anteriormente planteadas, pero teniendo en cuenta que es

necesaria la participación del cliente, la técnica de obtención de información al igual que en el levantamiento de requisitos será la entrevista con el cliente, esta técnica requiere de ser aplicada por una persona que tenga habilidades comunicativas además del trabajo en equipo y ser organizado, por lo tanto el planificador que será el encargado de esta actividad ha de contar con estas habilidades, atendiendo al tipo de software que se analiza es necesario que en para la descripción de ámbito participe el metodólogo que determine hasta donde llega el alcance educativo del software y se responsabilice por planificar la documentación didáctica.

2.5.2 Determinar la factibilidad del producto

Una vez identificado y descrito el ámbito del proyecto se conoce si el producto será factible, teniendo en cuenta cuáles serán las características y funciones finales que se entregará al cliente, así como conocer de la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señaladas en el ámbito.

2.5.3 Analizar los riesgos

El análisis de los riesgos contempla una serie de pasos que ayudan al equipo de proyecto a comprender y comportarse ante una situación no esperada, estos pasos se basan en la identificación de los riesgos, análisis de su probabilidad de ocurrencia y el daño que causará si ocurre, analizada esta información se clasifica según su probabilidad e impacto, finalmente se desarrolla un plan para gestionar los riesgos que más alta probabilidad de ocurrencia e impacto tengan. El plan de gestión de riesgos debe de ser lo más realista posible y actualizado. Para realizar la identificación de los riesgos es aconsejable realizar una lista de verificación donde se enfocan en algunos subconjuntos que permiten encontrar riesgos conocidos en categorías como:

Tamaño del producto: riesgos asociados con el tamaño del software que se desarrollará.

Características del cliente: riesgo asociado con las características comunicativas y experiencia del cliente.

Experiencia del personal: asociado a la experiencia técnica y global que debe tener todo el personal que desarrollará el producto.

Entorno de desarrollo: asociado a la disponibilidad de las herramientas y software necesarios para el desarrollo.

Una vez identificado los riesgos es necesario priorizarlos por probabilidad de ocurrencia, al desarrollar esto le permite al equipo asignar los recursos donde sea más necesario.

2.5.4 Definir los recursos requeridos

La cantidad de personas requeridas para el desarrollo de un proyecto de software solamente puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo personas mes o personas años) y seleccionar la posición dentro de la organización y la especialidad que desempeñará cada profesional, de igual forma pasa con los demás recursos necesarios para el software. Existen tres grandes categorías de recursos aplicables a cualquier proyecto.

- **Recursos Humanos**

Luego de haber analizado el ámbito el planificador será el encargado de gestionar los recursos humanos, teniendo en cuenta el esfuerzo que se requiere para el desarrollo del producto; los recursos humanos son definidos en dependencia del tamaño del producto a desarrollar, si el mismo es pequeño y se encuentran en una misma área de trabajo tiene que estar especificada la ubicación de ese recurso al igual que un producto con más recursos humanos y que no trabajen en la misma área.

Para la gestión de los recursos humanos es necesario hacer una especificación definiendo cuatro características fundamentales: la descripción del producto, la ubicación, su disponibilidad, cuándo se necesitará el producto y por cuánto tiempo.

Aunque la planificación varíe en dependencia de la complejidad, para el desarrollo de laboratorios virtuales es necesario un equipo metodológico que se encargue de la gestión de recursos educativos, el equipo metodológico debe estar compuesto por uno o más metodólogos en dependencia de qué tan complejo sea el sistema y de forma opcional un especialista en tecnología educativa que se encargue de desarrollar las distintas estrategias de intervención en el marco de las políticas educativas integrando los conocimientos específicos propios a las tecnologías aplicadas a la educación, debe de tener conocimientos para integrar herramientas informáticas con paradigmas de las ciencias de la educación; desarrollar proyectos tecnológicos dentro de la educación; y abordar las problemáticas del PEA.

- **Recursos de software reutilizables.**

La reutilización de componentes de software es una técnica que garantiza que una parte o la totalidad de un programa informático existente se puedan emplear en la construcción de otro programa. De esta forma se aprovecha el trabajo anterior, se economiza tiempo y se reduce la redundancia; en este tipo de software aunque se varía en las materias siguen una misma línea de trabajo, es práctico que se reutilicen componentes que no cambian independientemente del

contenido o el estilo del laboratorio. Siempre son necesarios los componentes de sonido, iluminación, carga de escenas y el reporte de los ejercicios en otros formatos.

- **Identificar los recursos del entorno.**

Los recursos del entorno están orientados al software y hardware necesario el planificador será el encargado de especificar cuál se necesita, el hardware proporciona una plataforma que soporta las herramientas (software) con que se producen los productos de trabajo basados en una buena práctica de la ingeniería de software.

Los laboratorios virtuales se pueden desarrollar en computadoras de medio rendimiento y cumplen sus objetivos, pero con el avance tecnológico y los constantes cambios en las tendencias de software para un mayor número de prestaciones sería necesario un hardware de más alto nivel, desarrollar laboratorios con el empleo de *joystick*³, permitiendo una mayor inmersión en el mundo virtual.

El trabajo del planificador se delimita a identificar cuáles serán esos recursos del entorno necesarios para el desarrollo y realizar una especificación de ellos.

2.5.5 Estimación.

De los sistemas basados en computadoras el software es el elemento más caro, la estimación del proyecto es una forma de resolver problemas un error puede hacer la diferencia o el beneficio, puede ser realizada aplicando técnicas.

- **Analógicas.**

Juicio de Expertos: Varios expertos valoran las condiciones del proyecto y estiman por separado. Se realiza luego un trabajo de mesa donde se llega a un consenso de las estimaciones producidas por los expertos, es aconsejable que se realice en proyectos que se encuentran en negocios y dominios desconocidos por la empresa o desarrolladores del proyecto que se ejecuta con expertos disponibles en otras empresas o en el mismo organismo.

Esta técnica no resulta ventajosa para el producto que se cuestiona, pues no siempre se cuenta con expertos en el tema en el cual está enfocado el desarrollo.

Analógica: Se estima por comparación con otros proyectos en el mismo dominio de aplicación y que hayan sido completados previamente. Las personas involucradas no sólo trabajan con su experiencia acumulada, sino que disponen también de datos de proyectos acabados relativamente similares al que hay que estimar. Así, por comparación, se pueden evaluar las diferencias entre el nuevo proyecto y los antiguos y

³ Palanca de mando.

extrapolar su coste. Además, se compararán las características de ambos proyectos (calidad del personal, formación, herramientas) para retocar la estimación.

▪ **Por descomposición.**

Del tamaño del software: La estimación de un proyecto solamente es tan buena como la estimación del trabajo para realizarlo, en tamaño es considerado el primer desafío del planificador, el término tamaño se refiere al resultado cuantificable del proyecto de software, el mismo se puede medir ya bien en líneas de código (LCD) o en puntos de función (PF), se utilizan en dos formas para estimar, una como variable de estimación para el tamaño de cada elemento de software y como métrica de líneas bases recolectadas a partir de proyectos previos y utilizados en conjunción de variables de estimación para realizar proyecciones de costo y esfuerzo.

Estimación basada en el proceso: El proceso se descompone en un conjunto relativamente pequeño de tareas y se estima el esfuerzo requerido para lograr cada tarea al igual que las estimaciones basadas en el problema, donde se realiza un bosquejo de las funciones del software obtenidas a partir del ámbito del proyecto. Una vez que se combinan estas las funciones con las actividades del proceso el planificador estima el esfuerzo.

Estas técnicas no son aconsejables por el hecho que resulta impredecible cuántas líneas de códigos tiene un laboratorio, aunque generalmente sus procesos tengan una base de desarrollo en común es poco probable que la cantidad de líneas de código sea un número invariable.

Estimación con casos de usos: Se estima tomando en consideración los casos de usos que se deben diseñar e implementar en el software.

Esta técnica es bastante utilizada, pero su principal inconveniente se basa en que es necesario que el desarrollo haya empezado para poder realizar la estimación y resulta que en ocasiones luego de haber empleado recursos para el inicio del desarrollo, el resultado de la planificación no sea factible y se hayan generado pérdidas en vez de ganancias. (15)

Luego de analizar algunas de las técnicas expuestas, **se recomienda la técnica basada en analogía**, porque resulta precisa en cuanto a la estimación real del proyecto teniendo como experiencias otros desarrollos que tributan a la misma área de estudio. Se puede realizar la estimación únicamente teniendo el conocimiento de cómo se comportó un desarrollo anterior. Un elemento importante es que no requiere que haya iniciado el desarrollo del producto, a diferencia de otras técnicas de estimación.

2.5.6 Definir estrategias de pruebas

Durante la etapa de planificación es importante establecer una buena estrategia de pruebas y seleccionar las técnicas adecuadas de estimación en función de los factores que afecten a las pruebas del proyecto. Esta estrategia de prueba ha de contar a parte de las pruebas que se realizan para comprobar el funcionamiento del software, con verificaciones para demostrar el cumplimiento de los aspectos educativos. Se realiza el Plan de Pruebas para definir el alcance, enfoque, recursos y planificación de las actividades de prueba.

2.5.7 Desarrollar un plan de proyecto

- **Establecer un conjunto de tareas significativo:** Se define un conjunto de tareas de trabajo de ingeniería de software, hitos y productos de trabajo que deben terminar un proyecto en particular.
- **Red de tareas:** Es una representación gráfica del flujo de tareas del proyecto, se utiliza como el mecanismo mediante el cual la secuencia y dependencia de tareas son la entrada de una herramienta automatizada y representan las principales tareas de la ingeniería. Se distribuyen las tareas a lo largo de la línea del tiempo del proyecto.
- **Cronograma:** Cuando se crea una calendarización del proyecto el planificador comienza con un conjunto de tareas, en caso de emplear herramientas automatizadas, el análisis del trabajo se introduce como una red de tareas, se introduce el esfuerzo, la duración y la fecha de inicio de cada una de las tareas, permitiendo asignar a un individuo en específico una tarea.
- **Herramientas para desarrollar cronograma:** El objetivo de las herramientas de calendarización de proyectos es permitir que un gestor defina las tareas de trabajo, establezca sus dependencias, asigne recursos a las tareas y desarrolle variedad de gráficos que evidencien el estado del proyecto, el uso de una herramienta en específico va de acuerdo a las comodidades que persiga el planificador, aunque se aconseja preferentemente que se use una herramienta de software libre.

2.5.8 Definir mecanismos de seguimiento del programa de trabajo

Una vez terminada la calendarización del proyecto es necesario realizar un seguimiento para comprobar el cumplimiento o no de las tareas de ingeniería propuestas a lo largo del proyecto. Una de las debilidades identificadas es la referente al exceso de reuniones en el equipo de proyecto, con el mecanismo de seguimiento, se debe de establecer la planificación para las mismas, evitando que estas sean innecesarias. El seguimiento que se propone para el mismo:

- Realizar reuniones semanales para comprobar el estado del proyecto donde cada uno de los miembros del proyecto comenten el avance de las tareas.
- En caso de conocer sobre la materia de la cual se realiza el LV, es necesario planificar talleres para llegar a soluciones.
- Determinar si se han cumplido los hitos del proyecto en las fechas establecidas.
- Comprobar si la fecha de inicio cumple con la fecha de fin pactada en la realización del cronograma.
- Reunirse con el cliente para conocer el grado de satisfacción con el avance del producto.
- Reunirse de ser posible con los futuros usuarios para conocer en qué medida se están cumpliendo los objetivos educativos del proyecto.

La planificación que es un componente principal de la gestión del proyecto de software, combina métodos de estimación y análisis de riesgos, que aunque la estimación nunca sea precisa representa una buena mezcla de datos y técnicas sistemáticas que pueden mejorar la precisión del software. En general establece caminos que guían para a tomar las mejores consideraciones y lograr la calidad requerida.

2.6 Fase de Modelado

Pressman define la actividad de modelado como la que permite la creación de modelos que posibilitan al desarrollador y al cliente entender mejor los requisitos del software y el diseño que logrará satisfacerlo. Esta actividad comprende dos acciones de la ingeniería de software, el análisis y el diseño.

El análisis abarca el conjunto de tareas de trabajo, tales como la investigación, elaboración, negociación, especificación, validación y gestión o administración de requisitos que conducen a elaborar el modelo de análisis.

El diseño abarca tareas de trabajo, diseño de datos/clases, diseño arquitectónico, diseño de interfaz, diseño al nivel de componentes que crean un modelo de diseño o especificación de diseño. A continuación se definen los elementos de la fase para los laboratorios virtuales con fines educativos.

2.6.1 Modelado del análisis

El modelado del análisis responde a una especificación de requisitos y a la representación completa del sistema que se construirá, este modelo constituye la primera representación técnica del sistema. El modelado del análisis es realizado por el analista del sistema a raíz de los requisitos posibilitados por el cliente.

El análisis de los requisitos permite generar la especificación de las características operacionales del sistema y establece las restricciones que debe de tener el software, le permite al analista desarrollar los modelos que representan los escenarios del usuario, las actividades, las clases, sus relaciones y el comportamiento de las mismas.

En un software educativo como en cualquier otro es importante el análisis para la validación de requisitos, el modelo del análisis representa dichos requisitos en diferentes dimensiones aumentando la probabilidad de encontrar errores, e inconsistencias. Ha de cumplir con tres objetivos fundamentales: describir lo que requiere el cliente, establecer una base para la creación del diseño del software y definir un conjunto de requisitos que puedan validarse una vez construido el software.

- **Enfoques del modelado del Análisis.**

Se definen dos enfoques para el modelado del análisis, el estructurado el cual considera al software como un transformador de información, este ayuda al ingeniero a identificar los objetos de datos, sus relaciones y la manera en la que estos objetos de datos se transforman mientras fluyen a través de las funciones de procesamiento del software.

El orientado a objeto examina un dominio del problema definido con un conjunto de casos de uso en un esfuerzo por extraer clases que definen el problema, donde cada clase tiene un conjunto de atributos y operaciones. Las clases se relacionan entre sí en formas diferentes y se modelan mediante diagramas de UML.

Para el desarrollo de los laboratorios virtuales con fines educativos, se selecciona el enfoque orientado a objeto que permite modelar el problema en casos de usos para ser convertidos posteriormente en clases con atributos y operaciones que permitan la reusabilidad, mantenibilidad, modificabilidad y fiabilidad.

Para llevar a cabo dicho enfoque Pressman define algunas tareas como:

- Deben de comunicarse los requisitos básicos del usuario entre el cliente y el ingeniero de software.
- Deben de identificarse las clases, así como sus atributos y métodos.
- Se define una jerarquía de clases.
- Se realiza una descripción del propósito de cada clase.

- **Modelado de datos.**

El modelado de datos es una de las actividades del análisis que define todos los objetos de datos⁴ que se procesan dentro del sistema y las relaciones entre los mismos, el objeto que se define es lógicamente en dependencia del problema que se modela. En la misma se tiene en

⁴ Representación de casi cualquier información compuesta que el software debe de entender.

cuenta el Diagrama Entidad Relación (DER), representando la pareja objeto- relación considerada la piedra angular del modelado de datos.

Esta actividad se tiene en cuenta porque aunque los laboratorios virtuales que se realizan actualmente en el proyecto y los desarrollos pasados no utilizan almacenamiento de datos, no tiene porqué ser así siempre, por lo tanto se considera a tener en cuenta en dependencia de la situación.

- **Elementos del modelo de análisis.**

Los requisitos de información, funcionales y de comportamiento se modelan mediante diferentes tipos de diagramas Pressman define cuatro elementos de los que a continuación se realiza una breve síntesis y posteriormente se selecciona cuáles son aplicables al desarrollo de LV con fines educativos.

Elementos basados en escenarios: representa el sistema desde el punto de vista del usuario.

- Casos de Uso.

- Diagramas de actividad.

- Diagramas de carril.

Elementos basados en clases: define objetos, atributos, operaciones y relaciones.

- Diagramas de clase.

- Paquetes de análisis.

- Modelos CRC.

- Diagrama de colaboración.

Elementos de comportamiento: presenta los estados del sistema así y sus clases, así como el impacto de los eventos sobre sus estados.

- Diagramas de estado.

- Diagramas de secuencia.

Elementos orientados al flujo: indica cómo se transforman los objetos de datos al realizarse las funciones de procesamiento.

- Diagramas de flujos de datos.

- Diagramas de flujos de control.

- Narrativas de procesamiento.

Analizando cada uno de estos elementos se descarta para el desarrollo de LV con fines educativos los elementos orientados al flujo, puesto que este modelo tiene una visión del tipo entrada-proceso-salida, donde se introducen datos que fluyen hacia el interior del sistema, se transforman y se muestran al exterior del sistema. Los LV se comportan sin un flujo constante

de datos, no intercambian comandos. Este elemento se enfoca en el flujo de objetos de datos, donde este concepto es propio del enfoque estructurado.

De los elementos basados en escenarios, se toman los casos de usos, pues permiten describir en más detalles las interacciones del usuario con el sistema. En los laboratorios virtuales se cuentan con numerosas actividades que serán más claras si el cliente las expresa mediante casos de usos, así describir los ejercicios mediante acciones del usuario y respuestas del sistema posibilita entender mejor el proceso. Además de permitir esbozar el problema desde un punto de vista más claro, en un lenguaje más entendible.

De los elementos basados en clases para los LV con fines educativos se toman los diagramas de clases, donde a raíz de los casos de usos se conforman las clases candidatas con sus atributos y operaciones. Los modelos clase-responsabilidad-colaborador (CRC) proporcionan un medio simple para identificar y organizar las clases relevantes para los requisitos del sistema o producto con el objetivo de lograr una representación organizada de las clases.

En los elementos de comportamiento se selecciona los diagramas de secuencias, que permiten representar el problema de una forma más dinámica y representar los estados de los mismos como un todo. Así con el uso de los diagramas de secuencias en los LV se puede evidenciar cómo interactúa un conjunto de objetos en la aplicación a través del tiempo.

2.6.2 Modelado del diseño

Una vez terminado el análisis de los requisitos comienza el diseño, siendo esta la última acción de ingeniería de la fase de modelado. El objetivo del diseño es aplicar un conjunto de prácticas, criterios y conceptos que conlleven a desarrollar un producto de alta calidad y que permitan implementar todos los requisitos de software además de evaluar la calidad del software antes de ser implementado. Sin diseño se corre el riesgo de desarrollar un sistema inestable. A diferencia del modelo de análisis presenta un nivel más bajo de abstracción.

Presenta el software a diferentes formas, primero se representa la arquitectura, después se modelan las interfaces que conectan el software con los usuarios finales, con otros sistemas y dispositivos y con los propios componentes que lo constituyen, por último se diseñan los componentes de software que se utilizarán en la construcción del sistema y en conjunto se obtendrá un modelo que abarca las representaciones arquitectónicas, de interfaz, en el nivel de componentes y de despliegue. El modelo del diseño es regido por una pirámide (*Ver figura 8*) que muestra las actividades fundamentales que se desarrollan en este modelo. A continuación se detallan cada una de estas actividades para los LV con fines educativos.

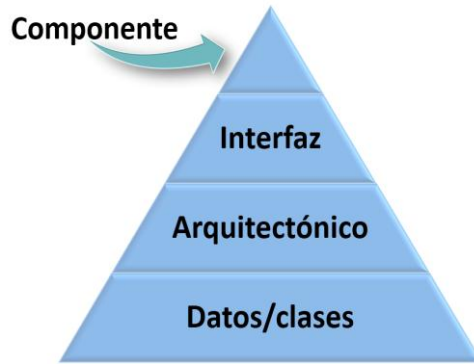


Figura 6. Pirámide de Diseño.

- **Elementos del modelado de datos/clases.**

El diseño de datos/clase es la primera actividad del diseño, la base de la pirámide, su realización para el desarrollo de los LV se basa en transformar los modelos del análisis y de clases en las clases del diseño y las estructuras de datos que se requieren para la posterior implementación del software.

Las clases y relaciones que se definieron en las tarjetas CRC, pertenecientes a los elementos del análisis basados en clases, así como los atributos y operaciones que proporcionan la base para el diseño de datos.

En pocas palabras es convertir las clases del análisis en clases del diseño con sus atributos y operaciones además de las estructuras de datos para ser implementadas.

- **Elementos del diseño arquitectónico.**

El diseño arquitectónico representa las estructuras de datos y los componentes del programa necesarios para construir el sistema, se selecciona el estilo arquitectónico, las estructuras y las propiedades de los componentes que constituyen el sistema, así como las relaciones entre los componentes arquitectónicos de dicho sistema. Los estilos arquitectónicos se analizan con el fin de obtener la estructura que mejor se ajuste a los requisitos obtenidos y a desarrollar un producto con la máxima calidad. Durante el diseño arquitectónico se crea un modelo de arquitectura que abarca la estructura del programa.

- **Estilos arquitectónicos.**

Los estilos arquitectónicos definen las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de software.

- **Patrones de arquitectura.**

Son varios los patrones de arquitectura que están definidos, los mismos expresan el esquema de organización estructural fundamental para sistemas de software. Para el desarrollo de los LV

con fines educativos se propone el uso de los patrones basado en capas y basado en componente.

- **Basado en capas.**

El patrón basado en capas funciona como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. Como desventaja tiene que los cambios en las capas de bajo nivel tienden a filtrarse hacia las de alto nivel, también se admite que la arquitectura en capas ayuda a controlar y encapsular aplicaciones complejas, pero complica no siempre razonablemente las aplicaciones simples.



Figura 7. Patrón basado en capas.

- **Basado en Componentes**

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado.

- **Patrones de diseño**

Un patrón de diseño es una descripción de clases y objetos que se comunican entre sí, adaptados para resolver un problema general de diseño en un contexto particular. Existen numerosos patrones de diseño, no es definible que patrón es o no recomendado para un laboratorio virtual, más bien se consideran buenas prácticas a tener en cuenta. Es por esta razón que estos patrones no son elementos puramente teóricos, sino que están fundados sobre una fuerte base práctica.

- **Descripción general de la arquitectura propuesta para LV con fines educativos**

Se recomienda el uso de una arquitectura basada en componentes combinada con una arquitectura en capas, el estilo en componente permite convertir los procesos que son comunes

en el desarrollo de laboratorios virtuales en componentes. Así, es posible ensamblar un cierto número de estos y lograr la funcionalidad básica de un laboratorio virtual a la vez que se reducen los costos de desarrollo. El estilo en capas permite separar por cada una de las capas los elementos del producto, en una capa de presentación, una lógica, una de soporte y una capa de elementos comunes. Dicha combinación de estilo permite construir una arquitectura, basada en la reutilización de los componentes e independencia en cada una de las capas.

Para más información esta combinación de estilos arquitectónicos es una propuesta de arquitectura desarrollada por el Ingeniero en Ciencias Informáticas Alejandro David Merzeau Martínez en su trabajo de diploma. (16)

La estructura general de un laboratorio cuenta con módulos que habitualmente no varían en sistemas similares, los mismos se encargan de áreas específicas dentro de la aplicación.

Módulos de pantallas de presentación y de gestión de menús. Representa las pantallas de presentación y despedida del programa, las pantallas de gestión de los menús principales, así como las áreas de trabajo.

Módulos de actividades interactivas. Contienen las diferentes actividades educativas que el programa puede presentar así como los ejercicios asociados a cada una de esas actividades.

Módulos de ayuda. Gestionan las ayudas. El acceso a la ayuda está en dependencia del tipo.

Ayudas sobre el funcionamiento del programa.

Ayudas didácticas, sobre los contenidos.

Módulos de evaluación. Gestionan el almacenamiento de información sobre el comportamiento cognoscitivo de los usuarios y la posterior generación de informes. La información que se desea mostrar en correspondencia a este comportamiento puede variar en dependencia de los parámetros que los encargados deseen medir.

Módulos auxiliares. Por ejemplo: gestión de posibles modificaciones de parámetros, utilidades para los alumnos (calculadora, diccionarios y otros).

El módulo de ayuda educativa permite el análisis de las respuestas de los alumnos siendo ésta una de las labores más difíciles y meticulosas de los diseñadores, ya que deben prever el mayor número posible de respuestas y además, tener prevista una salida para respuestas imprevistas.

▪ **Diseño de interfaz de Usuario**

El diseño de la interfaz de usuario es la actividad del diseño que comunica el hombre y el sistema. Esta actividad comienza con la identificación de los requisitos del usuario, de la tarea y del entorno. Una vez identificadas las tareas, se crean y se analizan los escenarios del usuario para definir el conjunto de objetos de acciones de la interfaz.

Esta actividad forma la base para la creación del formato en pantalla que representa el diseño gráfico y la colocación de los íconos, las posición del menú y la distribución de la información.

Theo Mandel en su libro crea tres reglas de oro para el diseño de la interfaz, que son perfectamente aplicables al diseño de interfaz de un laboratorio virtual.

- Dar el control al usuario.
- Reducir la carga de memoria del usuario.
- Construir una interfaz consecuente. (17)

▪ **Análisis de la interfaz de usuario para LV con fines educativos**

Para construir una interfaz se deben conocer las características de los usuarios finales, saber si son usuarios principiantes, o si son usuarios con un nivel de conocimientos sobre el tema, además de tener en cuenta la edad del público al que está destinada la aplicación. Para tener conocimiento del usuario se realiza una entrevista para comprender sus necesidades y sus motivaciones.

Una vez identificado el usuario es necesario conocer qué tareas realizará para conocer con cuáles objetos trabajará, o qué tareas se estarán realizando cuando el usuario esté trabajando, para obtener esta información se utiliza la técnica de casos de uso, en este caso se desarrolla para que el usuario exprese cuáles son las tareas que va a realizar, generalmente se escribe de forma informal. A partir del caso de uso se extraen las tareas, los objetos y el flujo de interacción y se define una jerarquía en las tareas que se desarrollan. Una vez terminado el análisis donde se han identificado con detalles todas las tareas que desarrollará el usuario final, comienza a realizarse el diseño.

▪ **Diseño de la interfaz de usuario para LV con fines educativos**

En el diseño de la interfaz se definen los objetos que esta tendrá y cómo funcionarán. Para esto se analizan los casos de uso descritos en el análisis. Una vez definidos los objetos se agrupan por tipo. En el momento que el diseñador está satisfecho con el diseño de un objeto y su acción se procede a realizar el formato en pantalla.

▪ **Formato de la pantalla**

Es una actividad del diseño de la interfaz del usuario, es el proceso donde se elabora el diseño gráfico y se colocan los iconos, se define el texto en pantalla, los nombres de las ventanas y se definen los elementos primarios y secundarios de los menús, el mapa de navegación del sistema.

El software con fin educativo que es el sistema que se analiza debe cautelar que la interfaz muestre aspectos identificables por el estudiante en el software, que permita darse cuenta del objetivo pedagógico que tiene los sistemas antes de interactuar con él.

En el diseño de la interfaz de usuario de un laboratorio virtual con fin educativo, es necesario tener en cuenta algunos parámetros como son:

- Definir diseño y organización de los menús horizontales y desplegados.
- Definir la iconografía y tipografía.
- Definir el tipo de navegación en el sistema virtual.
- Definir el diseño de las actividades a realizar.

Teniendo en cuenta siempre que es un software educativo y su objetivo primordial más que jugar es brindar contenidos y proporcionar conocimientos de forma general sobre alguna materia.

▪ **Patrones de interfaz de usuario**

Las interfaces de usuario han evolucionado tanto que se han creado patrones de interfaz para su utilización, entre los que se pueden recomendar para este tipo de software:

Navegación de alto nivel: Proporciona un menú de alto nivel, a menudo acoplado a un logotipo o una imagen que permite la navegación directa a cualquiera de las funciones básicas del sistema.

Indicador de programa: Proporciona una indicación de progreso cuando se realiza una operación.

▪ **Validación de la interfaz**

Una vez que se ha creado el prototipo de una interfaz es necesario evaluarlo para saber si las necesidades del usuario han sido satisfechas. Para ello se crea un prototipo de primer nivel donde el usuario lo evalúa, dicha evaluación por parte del usuario puede ser aplicando alguna técnica ya sea cuestionario, entrevista u otra que le propicie al diseñador datos para conocer el nivel de conformidad del usuario, se estudia esa evaluación y realiza las modificaciones pertinentes. Luego se vuelve a presentar el prototipo al usuario, para emitir un nuevo criterio, así cíclicamente hasta que se obtenga la interfaz esperada.

▪ **Elementos del diseño a nivel de componentes**

El diseño al nivel de componentes se presenta después que se ha completado la primera iteración del diseño arquitectónico, pues en esta etapa se ha establecido la estructura del programa. El diseño a nivel de componentes permite presentar el software de manera tal que se puede revisar que sea consistente y se pueden evaluar las estructuras, las interfaces y los algoritmos. Siguiendo el enfoque OO, el componente es el conjunto de clases que colaboran

entre sí, donde cada clase se ha elaborado completamente para incluir los atributos y operaciones relevantes para su implementación. El diseño a nivel de componente se obtiene a partir de tres fuentes, la información acerca del dominio de la aplicación, elementos del modelo de análisis, patrones y estilos arquitectónicos.

Los componentes en un software educativo, permiten ser reutilizados en productos similares. Y los que se define porque generalmente son usados en cualquier laboratorio son:

- Componentes para manejo de eventos.
- Componente para cargar las escenas.
- Componente para navegación.
- Componente de Sonido.
- Componente para el panel de herramientas.
- Componente de reportes en formato PDF.

Cada uno de estos componentes debe de ser independiente, desplegable y documentado.

2.7 Fase de Construcción

La fase de construcción es análoga al desarrollo donde se realiza la acumulación del conocimiento básico para construir el proyecto, o sea a la construcción de un sistema o producto. Se encarga de completar el análisis, diseño, desarrollo y prueba de toda la funcionalidad necesaria. El hito en esta fase culmina con el desarrollo del sistema con calidad de producción y la preparación para la entrega al equipo. Toda la funcionalidad debe haber sido implementada y las pruebas de la aplicación completadas. Como anteriormente se mencionaba, para estos sistemas, el análisis se realiza mediante el enfoque orientado a objeto, el Análisis Orientado a Objetos (AOO) y Diseño Orientado a Objetos (DOA) proporcionan una base bien definida para construir el software en Programación Orientada a Objetos (POO) con sus diversos requerimientos y optimizaciones. Además de tender a un enfoque reutilizable del software.

2.7.1 Implementación de código a cada clase a partir del Modelo de Diseño

Cada una de las clases y subclases identificadas en el diseño deben ser codificadas para proporcionar una interfaz que tenga relación con el usuario u otros sistemas.

Es necesario señalar que el uso de un estándar de codificación se propone en dependencia del equipo, no es objetivo establecer en esta sección un estándar específico, puesto que más que un elemento de la programación, es una buena práctica que deben de desarrollar los programadores para identificar en un momento determinado su código, son importante porque los estándares de codificación mejoran la lectura del software, permitiendo entender código

nuevo mucho más rápido y más a fondo, pero para que funcionen, cada persona del equipo de desarrollo deberá seguirlos al pie de la letra.

Aunque no se define ninguno en esta investigación se recomienda utilizar, el estándar de codificación definido por el Ing. Hassán Lombera Rodríguez para el Centro de Diseño y Simulación de Estructuras Mecánicas, que aunque son varios los estándares definidos, este constituye una guía más de cómo hacerlo y es por el cual actualmente se rige el Proyecto Laboratorios Virtuales. (18)

2.7.2 Reutilización

Cuando se realiza el desarrollo de laboratorios virtuales se hace factible la reutilización de unidades de software, aunque obviamente no sea el mismo producto el que se desarrolle siempre, los laboratorios virtuales tienen características en común que pueden ser aprovechadas por otros sistemas. La ingeniería basada en la reutilización es una aproximación del desarrollo que intenta maximizar la reutilización del software existente.

La reutilización de componentes por ejemplo, es muy bien vista por parte de los desarrolladores y no simplemente de componentes sino también de objetos y funciones; pueden reutilizarse componentes que ejecuten una sola función, como por ejemplo una función matemática. Una ventaja obvia de la reutilización es que disminuye el coste, pues se especifican, diseñan, implementan y validan menos componentes. Los componentes que se definen reutilizables se trataron anteriormente en este capítulo (*ver elementos del diseño a nivel de componentes*). Un ejemplo claro es el desarrollo de laboratorios que usan la misma arquitectura (anteriormente definida), cada uno con el objetivo de modelar disciplinas educativas diferentes, resultaría mejor que se definieran activos de software que cada uno pudiera utilizar en dependencia de su uso o no.

2.7.3 Pruebas

En este epígrafe se describen las principales tareas relacionadas con los procesos de validación y verificación utilizadas para la eliminación de errores, como son las pruebas y las revisiones. Cuando ya existe código ejecutable, se pueden realizar las pruebas del mismo. Las pruebas consisten en ejecutar el software con determinados datos de entrada y producir resultados que luego serán comparados con los teóricos. Se debe de asegurar que se han verificado todas las sentencias del mismo y que en las funciones externas se han realizado ensayos que aseguren que la entrada definida de datos produce los resultados que se esperan realmente. El objetivo de realizar las pruebas es descubrir los errores que no se encontraron en

etapas anteriores. Deben de empezar por lo pequeño y avanzar hacia lo grande, en otras palabras, se debe de realizar primero las pruebas a los componentes o a un grupo pequeño para probar su funcionamiento independiente y descubrir errores en las lógicas de los datos y del procesamiento encapsulado en el mismo, luego se procede a realizar pruebas de alto nivel para comprobar si las funcionalidades del sistema cumplen con los requisitos del cliente.

2.7.4 Estrategia de prueba para una arquitectura convencional

La estrategia de prueba proporciona una guía con los pasos que se darán como parte de la prueba, la misma es desarrollada por el jefe de proyecto, los ingenieros de software y los especialistas en pruebas. Describe el enfoque y los objetivos generales de las actividades de prueba. Incluye los niveles de prueba (unidad, integración y otros) a ser diseccionados y el tipo de prueba a ser ejecutadas (funcional, stress y otros.).

▪ Niveles de Prueba

El proceso de prueba es aplicado a diferentes escenarios o niveles de trabajo. Estos niveles a menudo se organizan por la secuencia en la que las porciones del sistema llegan a estar listas para probar. Cada fase por lo tanto cubre un objetivo de prueba a alto nivel.

Prueba de unidad: Se le realiza a los componentes más pequeños del software, para verificar que los flujos de control y de datos están cubiertos, tal y como se implementó en su código fuente para verificar que funcionen como se espera. Esta se recomienda por el uso de los componentes (antes definidos) en la construcción del laboratorio virtual.

Pruebas de integración: Es ejecutada para asegurar que los componentes en los que se basa la arquitectura propuesta operan correctamente cuando son combinados. Estas pruebas descubren en las especificaciones de las interfaces de los paquetes. Es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Pruebas de validación: Deben llevarse a cabo antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido

Prueba de sistema: Se prueba como un todo el software. Es la actividad de dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

▪ Pruebas no funcionales

Las pruebas no funcionales se realizan para verificar que el software desarrollado cumple con los requerimientos no funcionales establecidos. Para los LV con fines educativos se evalúan los requisitos no funcionales, específicamente de portabilidad, mantenimiento, fiabilidad, usabilidad y eficiencia.

Las pruebas de **Usabilidad**: Se utilizan para verificar la medida en la que un hardware, aplicación o software puede ser empleado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

Para el cumplimiento de la **Fiabilidad** se utilizan:

- Pruebas de Integridad: Enfocada a la valoración de la robustez (resistencia a fallos).
- Pruebas de Estructura: Enfocada a la valoración a la adherencia a su diseño y formación. Este tipo de prueba es realizada a las aplicaciones web asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano.
- Pruebas de estrés: Enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible)

Para el cumplimiento de la **Eficiencia** se utilizan:

- Pruebas de Benchmark: Es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido.
- Pruebas de Contención: Enfocada a la validación de las habilidades del elemento a probar para manejar aceptablemente la demanda de múltiples actores sobre un mismo recurso (registro de recursos, memoria)
- Pruebas de Picos: Se trata de observar el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga. Esta prueba se recomienda que sea realizada con un software automatizado que permita realizar cambios en el número de usuarios mientras que los administradores llevan un registro de los valores a ser monitoreados.

Para comprobar el cumplimiento del requisito no funcional de **Portabilidad** se usan las:

- Pruebas de Configuración: Enfocada a asegurar que el producto funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.
- Pruebas de Instalación: Enfocada a asegurar la instalación en diferentes configuraciones de hardware y software bajo diferentes condiciones (poco espacio en disco, insuficiente memoria RAM, entre otras)

De manera general se define para la portabilidad, mantenimiento y fiabilidad un procedimiento para que el probador tenga los indicadores para el cálculo de todas las métricas y evaluar esos requisitos que se proponen en la ISO⁵ 25 000. La eficiencia usa la herramienta automatizada *JMeter*⁶ y después de tener los resultados de esa herramienta se toman algunos valores para calcular las métricas de eficiencia y para usabilidad se propone una lista de chequeo de usabilidad a partir de las características que tienen los LV y referenciando el glosario de términos del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, para sus siglas en inglés).

Lo anterior analizado responde a la propuesta de pruebas que se argumenta en el **Procedimiento para realizar pruebas a requisitos no funcionales de laboratorios virtuales con fines educativos**, desarrollado por Dayanis Castellanos Rodríguez en su Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas. (19)

Está claro que estos laboratorios virtuales con fines educativos, entran dentro de la clasificación de un software educativo, parece perfecto realizar pruebas como las antes descritas, pero hay que tener en cuenta que estas pruebas sólo comprueban el estado funcional del software, la parte computacional y la interacción hombre-máquina, pero teniendo en cuenta el fin educativo que tienen estos sistemas, no es aún frecuente la participación de un evaluador educativo.

▪ Evaluación del laboratorio virtual

Evaluar un laboratorio virtual con fin educativo establece como finalidad, orientar el uso pedagógicamente adecuado, en otras palabras, conocer si se cumplieron los objetivos pedagógicos y educativos con los cuales fue concebido el producto. Puede surgir entonces la pregunta ¿qué evaluar? Algunos de los aspectos siguientes:

Usabilidad del programa: Medida en que el sistema es fácil de aprender y de utilizar. Teniendo en cuenta los siguientes aspectos.

Facilidad del aprendizaje: Es la medida en la que un usuario de cualquier nivel⁷ comprende cómo utilizar inicialmente el sistema y cómo a partir de esa utilización llega a un máximo nivel de conocimientos con la aplicación. Este elemento puede formar indicadores predictivos, donde los conocimientos adquiridos por los estudiantes, resultan suficientes para conocer los resultados de futuras interacciones.

⁵ Organización Internacional de Normalización (ISO)

⁶ herramienta libre y escrita en Java para desarrollar pruebas de carga y rendimiento.

⁷ Bajo: poco conocimiento del programa.

Medio: conocimiento del programa medianamente.

Ato: conocimientos avanzados.

Flexibilidad: Es la forma en que el usuario intercambia información con el laboratorio y viceversa, permitiendo la adaptabilidad del mismo a todos los tipos de usuarios y la migración de tareas, para realizar transferencias de control, ya sea del usuario al sistema o del sistema al usuario.

Recuperabilidad: Comportamiento del sistema al usuario al tener la posibilidad de corregir una acción una vez que ha reconocido un error.

Disponibilidad de ayuda: Posibilidad que brinda el laboratorio de mostrar la ayuda en cada una de las áreas de trabajo, sin tener que ir a un área en específico y pausar la práctica. Además debe de mostrar la ayuda con texto comprensivo, de tamaño adecuado y tipología formal y no puede ser obstructiva (que no impida el uso normal del laboratorio).

También es necesario evaluar el producto de forma pedagógica. Con parámetros como:

Contenido general: Si se adecúa el contenido con el objetivo pedagógico del mismo. Evaluando cantidad y calidad de la información, tratando por todos los medios, que sea información actual y referenciada.

Contenido pedagógico: Se encarga de determinar cómo se adecúan pedagógicamente los objetivos, las intenciones formativas. Evaluar si es necesario que los usuarios tengan o no conocimientos previos de la aplicación para operar con el laboratorio. Y si el contenido logra desarrollar habilidades con las actividades.

Nivel de aprendizaje: Comprueba si el laboratorio se encuentra en el nivel de aprendizaje de los usuarios para los cuales fue concebido el producto.

Intuitivo: Comprobar si el sistema fue diseñado para ser utilizado con la presencia del profesor o si solamente es capaz de impartir los conocimientos.

Variedad de actividades: El laboratorio ofrece variedad de actividades graduadas en varios niveles de dificultad que permiten al alumno escoger la que desea.

Para responder cada uno de estos parámetros se realiza una prueba de campo, antes de editar la versión definitiva, se trata de realizar una prueba en situación real o muy similar con el fin de detectar problemas en cuanto a objetivos, características pedagógicas y que puedan ser corregidos. Para recoger dicha información en cuanto al comportamiento de los parámetros antes expuestos, se puede utilizar una guía de evaluación educativa (*Ver Anexo 3*), que es desarrollada por el evaluador educativo. La planilla va en dependencia del software a evaluar y sus objetivos.

2.8 Fase de Despliegue

En muchas ocasiones los desarrolladores cuando terminan la fase de construcción piensan que han terminado el producto, pero para ello es importante realizar un despliegue exitoso. El despliegue comienza cuando el código ha sido suficientemente probado, ha sido aprobado para su liberación y ha sido distribuido en el entorno de producción. Una vez terminada la fase de construcción del software y por ende haber realizado las pruebas pertinentes, se tiene listo el producto para hacer entrega al cliente. Pressman define dicha fase como, la fase que sugiere que al entregar el software al cliente este lo evalúa y proporciona información a partir de su evaluación. En general el planeamiento de despliegue de sistema requiere un alto grado de colaboración y preparación del cliente. Una conclusión exitosa para un proyecto de software puede ser impactada severamente por factores fuera de alcance del desarrollo de software como los usuarios que están mal preparados para un cambio al nuevo sistema.

El producto está terminado, ahora es el cliente el encargado de evaluarlo y mostrar su conformidad o no con el mismo, para ello se planifica un encuentro en el que el cliente se encarga de operar con el software en busca de alguna deficiencia. Se debe entregar el producto debidamente empacado, etiquetado y con información sobre su contenido, aplicación, población objetivo y requerimientos de instalación

2.8.1 Documentación del producto

La documentación del programa se debe de hacer con tanto cuidado como el mismo producto informático, ya que constituye un elemento indispensable para que los usuarios puedan obtener el máximo rendimiento de las prestaciones que ofrece el material. La documentación externa del programa se refiere a todo el material confeccionado desde las etapas iniciales del desarrollo que incluye los diagramas que muestran el flujo de proceso, el diseño de los módulos, éstos permiten conocer el funcionamiento del programa y realizar alguna actualización del mismo en caso de ser necesario. Se recoge a forma de expediente de proyecto, en el que debe encontrarse toda la documentación generada por el desarrollo del producto y de forma organizada, se debe de documentar cuanto sea necesario para el entendimiento posterior ya sea por parte del cliente o por el equipo de desarrollo.

- **Manual del usuario.** Debe de explicar todo lo que necesita saber un usuario del programa, modo de uso, la forma de desenvolverse en el entorno para utilizarlo sin problemas y sacar el máximo partido de sus posibilidades.

- **Guía didáctica.** Está dirigida a los profesores. Ofrece sugerencias sobre la integración curricular del programa, sus formas de uso, actividades complementarias, estrategias para evaluar el rendimiento de las situaciones educativas que genera el programa, u otros.
- **Manual de instalación.** Debe de brindar la información de los requerimientos no funcionales para el correcto funcionamiento del producto así como, elementos de hardware y software en caso de ser necesarios y mostrar al usuario de forma entendible el procedimiento a seguir para la instalación.

2.8.2 Entrenamiento a usuarios finales

Se hace necesario definir una planificación para el entrenamiento a los usuarios finales que interactuarán con el producto. En el entrenamiento se brindarán todos los procedimientos de instalación teniendo en cuenta el hardware o software necesario para que funcione la aplicación así como el modo de operación con la aplicación para lograr una retroalimentación. El entrenamiento se ha de tener en cuenta desde que se realiza la planificación inicial del proyecto

2.8.3 Aceptación del Usuario

Es recomendable realizar pruebas con los usuarios finales y comprobar su grado de satisfacción y efectividad del software, pedirles que emitan su opinión acerca del producto, para tener referencias a la hora de realizar alguna otra versión o bien otro producto similar.

2.8.4 Consideraciones parciales

En este capítulo se propone la solución al problema planteado.

- Se desarrolla un análisis estratégico del proyecto Laboratorios Virtuales, el cual se ha escogido como escenario a valorar para desarrollar y aplicar esta investigación.
- Se especifican cada una de las actividades para el desarrollo de LV con fines educativos.
- Por cada una de estas actividades se detalló un conjunto de tareas que se deben de realizar para dar solidez a cada una de las actividades.
- Se especificó el artefacto de corte educativo guión didáctico.
- Se definió el rol de metodólogo para el levantamiento de requisitos con sus habilidades y responsabilidades.
- Se mostró una forma de evaluar los aspectos didácticos presentes en el software teniendo en cuenta los indicadores especificados.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA.

3.1 Introducción

Actualmente, en el proyecto productivo Laboratorios Virtuales se desarrollan 9 laboratorios virtuales, el propósito de este capítulo es realizar mediante una observación el comportamiento en el proceso de desarrollo de los laboratorios virtuales sin utilizar el marco de trabajo propuesto y valorando la utilización de este marco desarrollando un prototipo que valide parte de la propuesta, haciendo uso de elementos propuestos en cada una de las fases del marco, teniendo en cuenta que no todas las fases pueden ser aplicadas al prototipo y observando algunos indicadores en cuanto a la organización de la información, las actividades de los roles, los artefactos necesarios u otros elementos. Luego de realizar dicha observación se realiza un análisis comparativo de ambas situaciones.

3.2 Presentación del prototipo

El objetivo del prototipo es ser una versión operativa preliminar del sistema para fines de demostración y evaluación. El prototipo se desarrolla con el objetivo de llevar a cabo su desarrollo bajo el uso de aspectos que se proponen en el marco en cada una de las fases y compararlo con el desarrollo de laboratorios que no usan el marco; para así demostrar que el marco resulta aplicable y beneficia al proceso de desarrollo.

En el desarrollo del mismo se utilizaron los componentes que se mencionaron con anterioridad, así como la presencia de módulos con los que debe de contar normalmente un laboratorio.

3.2.1 Vistas del prototipo desarrollado

A continuación se muestran imágenes de las diferentes ventanas del prototipo.



Figura 8. Vista de inicio.

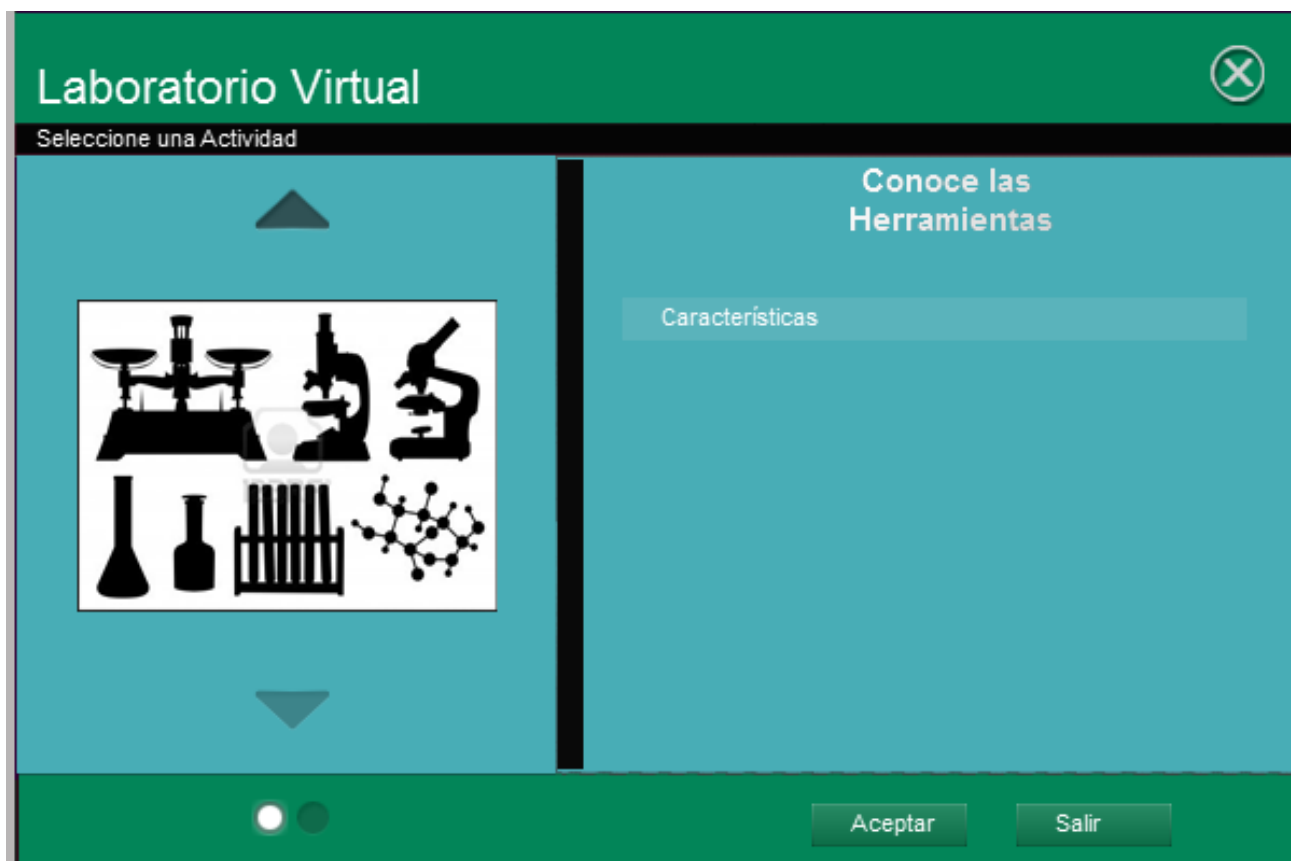


Figura 9. Vista de selección de las actividades.

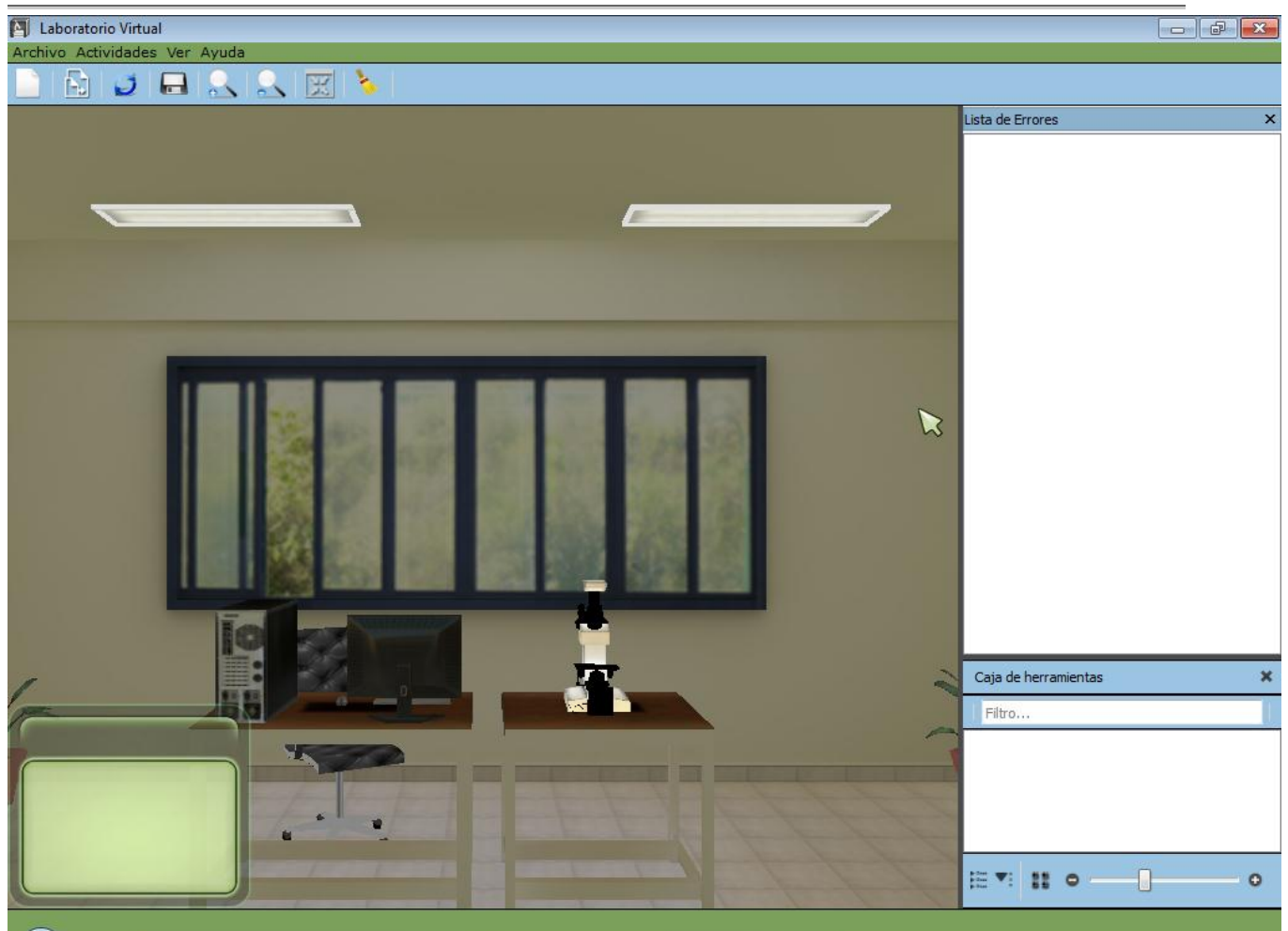


Figura 8. Vista del área de práctica del laboratorio.

3.3 Análisis y discusión de la fase de Comunicación

Como se fundamentó en la fase de comunicación descrita en el capítulo 2, la actividad que más importancia tiene en esta etapa es llevar a cabo una buena ingeniería de requisitos. A continuación para esta fase se describen elementos que se realizan en el desarrollo actual de los LV como son:

- El levantamiento de requisitos es desarrollado por el analista de sistemas y en ocasiones hasta otros roles que no desempeñan esta actividad (desarrolladores).
- No se cuenta con especificaciones de requisitos educativos.
- No se especifican concretamente los elementos para el diseño de los modelos del laboratorio.
- Se emplea como técnica de requisito la entrevista.

Estas afirmaciones se obtuvieron observando el proceso de desarrollo de los laboratorios virtuales actuales.

Analizando los puntos expuestos con anterioridad es apreciable que con el uso del marco se pulen estas deficiencias al proponer al analista y al metodólogo como roles activos en el levantamiento de requisitos, la utilización de los artefactos: Planilla de diseño educativo y Planilla para la solicitud de diseño, además de mantener la técnica de entrevista pero más específica la variante entrevista semiestructurada que permite mayor flexibilidad en consideración a la anterior.

3.4 Análisis y discusión de la fase de Planeación

Esta fase no ha sido probada en el prototipo pero sí se analiza en el desarrollo actual y como se realizaría de utilizarse el marco. Aunque no varía en gran ángulo. A continuación elementos del desarrollo actual de los LV:

- La técnica de estimación requiere haber iniciado el desarrollo, que implica que en caso de no ser factible se hayan utilizado recursos.
- No se tiene conocimiento de la planificación de la documentación didáctica, porque no se tienen definidos estos artefactos.

El marco propone tareas generales que acotan las responsabilidades del planificador, permitiendo desarrollar un plan de proyecto que cumpla las expectativas, además de poner bajo la responsabilidad del metodólogo la entrega de toda la documentación didáctica, así como el seguimiento y control de las tareas de cada uno de los roles, se propone como técnica de estimación a usar la Analógica, bajo la justificación de ser la técnica más precisa, pues permite realizar la estimación del proyecto antes de que comience el desarrollo del mismo, solamente basándose en experiencias de desarrollos pasados.

3.5 Análisis y discusión de la fase de Modelado

En el desarrollo actual esta fase no varía en gran medida con lo que se propone en el marco y simplemente se han utilizado algunos elementos en el desarrollo del prototipo como la arquitectura y la presencia de los módulos generales.

En el marco se propone el enfoque de análisis por el que se debe de regir para desarrollar los LV; analiza los elementos que se utilizan del modelo de análisis (basado en clases, basado en escenario, de comportamiento). Considera el modelado de datos en caso que el desarrollo de LV que utilice base de datos. Define las actividades del diseño a tener en cuenta en su caso

especial la actividad de formato en pantalla, sumamente importante para desarrollar la interfaz de usuario, presentando iconografía, tipografía y otros elementos. Explica la validación de esta interfaz mediante la presentación al usuario de un prototipo con las características finales que mostrará el producto una vez finalizado su desarrollo.

3.6 Análisis y discusión de la fase de Construcción

Se observó que la construcción de un laboratorio virtual se ha desarrollado actualmente de la misma forma como cualquier otro tipo de software siguiendo el flujo de las actividades de ingeniería en cuanto a la implementación, pero se hacen apuntes como:

- La codificación de dichas clases aunque se realice de la manera más clara y entendible para los implicados no se rige por ningún estándar de codificación.
- En la implementación se hace uso del concepto de reutilización, usando componentes reutilizables en el desarrollo de cada uno de los laboratorios.
- No se realizan evaluaciones para comprobar el cumplimiento de los aspectos educativos.

El uso del marco permite el uso de un estándar de codificación el cual se referencia, propone los componentes reutilizables en el desarrollo de los laboratorios virtuales con fines educativos y propone de una forma sencilla cómo realizar evaluaciones para comprobar los aspectos educativos.

3.7 Análisis y discusión de la fase de Despliegue

El despliegue se realiza en el ambiente destinado por el cliente y se tiene la experiencia donde se afirma que

- Solamente se le hace entrega al cliente del producto, su código fuente y el manual del usuario.

Analizando la propuesta que realiza el marco de trabajo, el mismo propone que se le haga entrega de una guía didáctica para que guíe al futuro usuario en la práctica de las actividades y le permita evaluar las situaciones educativas que genera el programa, acompañado del manual de instalación y el mismo manual de usuario que se le ha hecho entrega hasta el momento, realizando una observación de cómo se realiza la actividad real, se puede apreciar que desarrollar el despliegue de un laboratorio utilizando el marco permite brindar más información al cliente y por lo tanto se logre una mayor satisfacción de su parte.

3.8 Valoración general de los resultados

Atendiendo a la observación que se ha realizado sobre el desarrollo de este tipo de producto y usando elementos del marco luego de desarrollar el prototipo se pueden establecer algunas comparaciones como las que se recogen en la tabla que se muestra a continuación:

Basado en la experiencia actual.	Basado en el uso del marco de trabajo.
Deficiencias en el levantamiento y la especificación de requisitos.	El levantamiento y la especificación de requisitos son realizados por el analista y el metodólogo.
Los diseñadores no modelan los elementos del laboratorio acorde a lo exigido por el cliente	Se realiza la planilla de solicitud de diseño con las características que debe tener cada elemento del LV.
Se desarrolla el producto guiado por la metodología RUP que no recoge información educativa.	El marco posibilita ser flexible en el uso de una metodología.
No se tiene como almacenar los requisitos didácticos.	Propone la planilla de diseño didáctico.
Utiliza una técnica de estimación menos factible.	Propone una técnica de estimación que resulta más precisa para el desarrollo de LV con fines educativos.
Utiliza la entrevista como técnica de levantamiento de requisitos.	Utiliza la entrevista y dentro de esta las semiestructuradas como técnica de levantamiento de requisitos, que permite obtener más información.

Tabla 3. Comparación de los resultados.

En el desarrollo del prototipo no se pudo emplear el marco en su totalidad, pues resulta imposible ser aplicado a simplemente una parte de lo que sería en su final un producto completo, pero teniendo esa experiencia y la observación al proceso de desarrollo de un laboratorio real, se pudo realizar la tabla que se mostró anteriormente.

Al analizar la tabla y enfocarse en lo que plantea el problema estratégico definido al inicio de la investigación que corrobora el problema en la estructura del proceso de desarrollo de los

laboratorios virtuales, es posible una vez terminado la concepción del marco arribar a una conclusión importante para esta investigación: Utilizar el marco de trabajo en el desarrollo de un laboratorio virtual permite lograr una mejor organización en el proceso de desarrollo del software, generar la documentación que se necesita, definir los roles encargados de los procesos educativos con sus habilidades y responsabilidades, facilitar el trabajo del diseñador al modelar los objetos, estructurar el proceso de desarrollo de software y definir cada una de las actividades específicas en el desarrollo de los LV con fines educativos

3.9 Consideraciones parciales

En este capítulo se estableció una comparación entre la producción de LV que se desarrollan actualmente y cómo se comportarían de usarse el marco, el cual fue parcialmente aplicado a un prototipo no funcional.

- Se permitió comprobar que el marco mejora el proceso de desarrollo de software al perfeccionar sus actividades.

CONCLUSIONES GENERALES

Con la realización de este trabajo, se desarrolló un marco de trabajo ingenieril que permite mejorar la organización del proceso de desarrollo de software de laboratorios virtuales con fines educativos, dando cumplimiento al objetivo planteado al inicio de la investigación, teniendo en cuenta también que:

- Se logró especificar en cinco fases fundamentales el desarrollo de un laboratorio virtual con fin educativo, describiendo cada una.
- Se confeccionó el Guión Didáctico artefacto de corte educativo, para mejor desarrollo del producto, que permiten documentar la información pedagógica necesaria.
- Se confeccionó la planilla de Solicitud para el diseño, que le facilita al diseñador el proceso de modelado de los objetos.
- Se propuso el rol de metodólogo con sus habilidades y responsabilidades, encargado de atender los aspectos didácticos – pedagógicos del software.
- Se comprobó mediante la comparación del marco con el desarrollo actual de los LV, que el marco de trabajo propuesto ofrece mejoras al proceso de desarrollo.
- Se obtuvo la conformidad del marco de trabajo por parte del equipo de proyecto de Laboratorios Virtuales mediante una Carta de Aceptación, que expresa el aval y la consideración del uso del marco para futuros desarrollos. (*Ver Anexos*)

RECOMENDACIONES

- Utilizar el marco de trabajo propuesto en posteriores desarrollos de laboratorios virtuales.
- Hacer extensible la propuesta del marco hasta la especificación de las actividades sombrillas.

BIBLIOGRAFÍA

1. **Bauer.**Software Engineering. Amsterdam : s.n., 1972.
2. **Bhoem.**Software Engineering. 1976.
3. **Pressman, Roger S.**Ingeniería del Software: Un enfoque práctico 6ta edición. 2005
4. **Institute, Project Management.** Guía de los Fundamentos para la Dirección de Proyectos. 2008 .
5. **Amparo Navasa Martínez.** Marco de trabajo para el desarrollo de arquitecturas software orientado a Aspectos. Cáceres : Universidad de Extremadura, 2008.
6. **Ramos, Javier Saldaña.** Un Marco para la Mejora en la Gestión de los Equipos de Desarrollo Software Global. Leganés : s.n., 2010.
7. **Echeagaray, Dra. Alicia Reyes.** Tendencias actuales para el desarrollo de software. s.l. : Universidad Nacional Autónoma de México.
8. **Marques., Pere.**El software educativo. . Universidad Autónoma de Barcelona. : s.n., 1996.
9. **Vary, James.**Informe de la reunion de expertos sobre laboratorios virtuales. 1999.
10. **Tomasini, Alfredo Aclé.** Planeación estratégica y control total de calidad. Mexico : Grijalbo, 1989.
11. **Brooks, Fred.**The Mythical Man-Month. 1975
12. **Mejora, Programa de.** 0516_Roles y Responsabilidades. s.l. : Universidad de las Ciencias Informáticas.
13. **Cataldi, Zulma.** Extención para la creación de software educativos. Buenos Aires : s.n.
14. **Venezuela., Universidad Bolivariana de.** Elicitacion de Requisitos Material Complementario. 2010.
15. **Proyectum.** [En línea] 8 de noviembre de 2010. [Citado el: 5 de Mayo de 2013.] <http://proyectum.wordpress.com/2010/11/08/tecnicas-de-estimacion/>.
16. **Martínez, Alejandro David Merzeau.** Arquitectura de software para los Laboratorios Virtuales. La Habana : Universidad de las Ciencias Informáticas, 2012.
17. **Mandel, Theo.** The Elements of User Interface Design s.l. : Wiley, 1997.
18. **Rodríguez, Ing. Hassán Lombera.** Estándar de codificación para el lenguaje C++ utilizado en el Centro de Diseño y simulacion de piezas mecánicas. La Habana : s.n., 2011. 1.

19. **Rodríguez., Dayanis Castellanos.** Procedimiento de pruebas no funcionales para laboratorios virtuales con fines educativos. La Habana : Universidad de las Ciencias Informáticas., 2013.
20. **blogspot.** [En línea] [Citado el: 9 de diciembre de 2012.] <http://klcjw10.blogspot.com/2010/05/marco-de-trabajo.html>.
21. **Dijkstra, Edsger.** "Notes on Structured Programming". 1972.
22. **Figuroa, Antonieta Abud.** Ingeniería de Software Educativo.. 1, Orizaba, Veracruz, México : s.n., 2009, Revista Internacional de Educación en Ingeniería, Vol. 2.
23. **Sommerville, Ian.** Ingeniería del Software. España : Pearson Education SA, 2005, Vol. 7ma edición.
24. **Software, Laboratorio Nacional de Calidad del.** GUÍA DE VALIDACIÓN Y VERIFICACIÓN. España : Instituto Nacional de Tecnologías de la Comunicación, S.A., 2009.
25. **Velásquez, Cesáreo Morales** .Evaluación de software educativo.. México : ILCE, 1998.
26. **Castañón., Miguel Ángel González.** Orientaciones del uso pedagógico del software educativo.. Colombia : s.n., 2009.
27. **Panqueva., Alvaro Hernan Galvis.** Ingeniería de software educativo. Colombia. : s.n., 2001.
28. **eumed.** [En línea] [Citado el: 2 de Mayo de 2013.] http://www.eumed.net/libros-gratis/2012b/1204/capacitacion_evaluacion.html.
29. **MeRinde.** [En línea] [Citado el: 2 de Mayo de 2013.] http://merinde.net/index.php?option=com_content&task=view&id=41&Itemid=159.
30. **Eduotec.** [En línea] [Citado el: 2 de Mayo de 2013.] <http://edutec.rediris.es/Revelec2/presentacion.html>.

ANEXOS

ANEXO 1

**SOLICITUD DE DISEÑO
PARA LABORATORIOS VIRTUALES
CON FINES EDUCATIVOS
<NOMBRE DEL PRODUCTO>**

Control del Documento

Título:

Versión:

Nombre	Cargo
--------	-------

Reglas de Confidencialidad

Clasificación: USO INTERNO

Forma de distribución: PDF Digital

Control de Cambios

Versión	Sección, Figura, Tabla	Tipo A, B, M	Fecha	Autor del cambio	Descripción del Cambio

Introducción

[Se describe el alcance del documento y dando una breve explicación o resumen del mismo. Debe permitir poder hacerse una idea sobre el contenido del texto]

Solicitud de diseño.				
__2D __3D: <i>[Seleccionar dimensiones del objeto]</i>				
Nombre del Laboratorio: <i>[Se especifica el nombre del producto]</i>				
Especificaciones generales: <i>[Se describen características comunes para todos los objetos de forma general]</i>				
Actividad # Nombre de la actividad: <i>[Se especifica el número de la actividad a la que corresponden los objetos y el nombre de la actividad]</i>				
Nombre Objeto.	Identificador.	Dimensiones	Colores	Concepto.
<i>[Se escribe el nombre del objeto correspondiente]</i>	<i>[Se determina un identificador para el objeto con el que será reconocido cuando se modele]</i>	<i>[El tamaño que debe de tener ese objeto, en caso de que sea un esbozo realizado por un artista el mismo debe especificar estas dimensiones, si son necesarias para modelar el objeto]</i>	<i>[Se determinan los colores que debe de tener el objeto y de ser posible en que partes se encuentran los mismo]</i>	<i>[Se refiere a una información visual que puede ser una imagen o un esbozo que realice un artista]</i>

GUIÓN DIDÁCTICO PARA LABORATORIOS VIRTUALES CON FINES EDUCATIVOS <NOMBRE DEL PRODUCTO>

Control del Documento

Título:

Versión:

Nombre	Cargo
--------	-------

Reglas de Confidencialidad

Clasificación: USO INTERNO

Forma de distribución: PDF Digital

Control de Cambios

Versión	Sección, Figura, Tabla	Tipo A, B, M	Fecha	Autor del cambio	Descripción del Cambio

Introducción

[Se describe el alcance del documento, dando una breve explicación o resumen del mismo. Debe expresar una idea sobre el contenido del texto]

Objetivo general del laboratorio

[Constituyen todas las metas que se desean alcanzar]

Objetivos didácticos

[Señalan las metas hacia donde encaminar la actividad escolar. Deben de estar planteado de forma clara y ser lo suficientemente flexible para poder adaptarse al nivel de aprendizaje de cada estudiante]

Objetivos generales

[Constituyen todas las metas que se desean alcanzar]

Uso del medio

[Debe referirse al uso educativo que se le dará al software, enfocarse al medio en que será utilizado]

Público al que está destinado

[Debe estar enfocado a todos los estudiantes que inicien una formación académica determinada]

Contenido

[Temas específicos que se tratarán en las prácticas del laboratorio atendiendo a la materia de la cual se desarrolla el laboratorio]

Nombre de la actividad: <i>[nombre de la actividad]</i>			
Objetivo de la actividad: <i>[objetivo que cumple esa actividad]</i>			
Tiempo estimado: <i>[tiempo que demora en realizar la práctica]</i>			
Actividades mentales a desarrollar: <i>[actividades intelectuales que hay que suscitar en el alumno para que alcance los objetivos de una manera duradera]</i>			
Ejercicio <i>[Nombre del ejercicio.]</i>	Pasos metodológicos. <i>[Secuencia en la que se deben de realizar las operaciones para dar solución al ejercicio]</i>	Descripción de la actividad <i>[Descripción más detallada de la actividad y el entorno en el que se desarrolla]</i>	Elementos que se utilizan. <i>[Se describen los</i>

			<i>elementos principales que se utilizan en la práctica del laboratorio, las herramientas con los que se trabajan]</i>
Pautas cromáticas. <i>[Colores generales del laboratorio]</i>		Tipografía. <i>[Fuente que se determina para el laboratorio]</i>	

ANEXO 3

Propuesta de una guía para realizar la evaluación del laboratorio en dependencia de los aspectos que se evalúan.

Nombre del producto: <i>[nombre del producto]</i>	
Público al que está destinado: <i>[estudiantes universitario, primarios, otros niveles]</i>	
Nivel del educando: <i>[nivel de conocimiento de la aplicación (bajo, medio y alto)]</i>	
Aspectos pedagógicos a evaluar	Resultados
<i>[se precisan los elementos que se desean evaluar del software, se puede hacer en forma de preguntas para hacer menos engorroso el proceso]</i>	<i>[se colocan los valores en correspondencia, sí, no o parcialmente, puede ir acompañado de una descripción en caso de que lo considere necesario]</i>

La Habana, junio de 2013

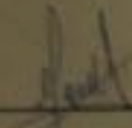
De mi consideración:

Por este medio se expresa la conformidad y aceptación del Marco de trabajo ingenieril para el desarrollo de Laboratorios Virtuales con fines educativos, que tributa al Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas, de la autora Yamika de la Ciudad Caceres Mayo y se le reconoce el aporte que representa para el proyecto productivo Laboratorios Virtuales perteneciente al Centro de Informática Industrial de la Universidad de las Ciencias Informáticas.

El marco de trabajo soluciona el problema existente en cuanto a los aspectos pedagógicos y didácticos que no son contemplados en el proceso de desarrollo actual. Organiza el proceso de desarrollo en cinco fases fundamentales, describiendo para las mismas las actividades asociadas y en casos necesarios los roles que intervienen, detallando sus habilidades y responsabilidades. Hace la propuesta de una técnica de estimación más factible. Y propone una mejor especificación de los elementos 2D y 3D que simulan el entorno real.

Por lo tanto: El equipo de proyecto está satisfecho con la propuesta de Marco de trabajo ingenieril para el desarrollo de laboratorios virtuales con fines educativos, lo avala y acepta, considerando su uso para posteriores desarrollos.

Para que así conste, se firma a los 3 días del mes de Junio del 2013.


Yaself Machado Tugores.

Jefe de Proyecto Laboratorios Virtuales.

