



Universidad de las Ciencias Informáticas

Facultad 8

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Una búsqueda de elementos arquitectónicos
del software educativo en los proyectos
productivos del MINED**

Autor: Yalieski Revé Borges

Tutor: Leonardo San Román Labaut

Ciudad de la Habana, junio del 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 8 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

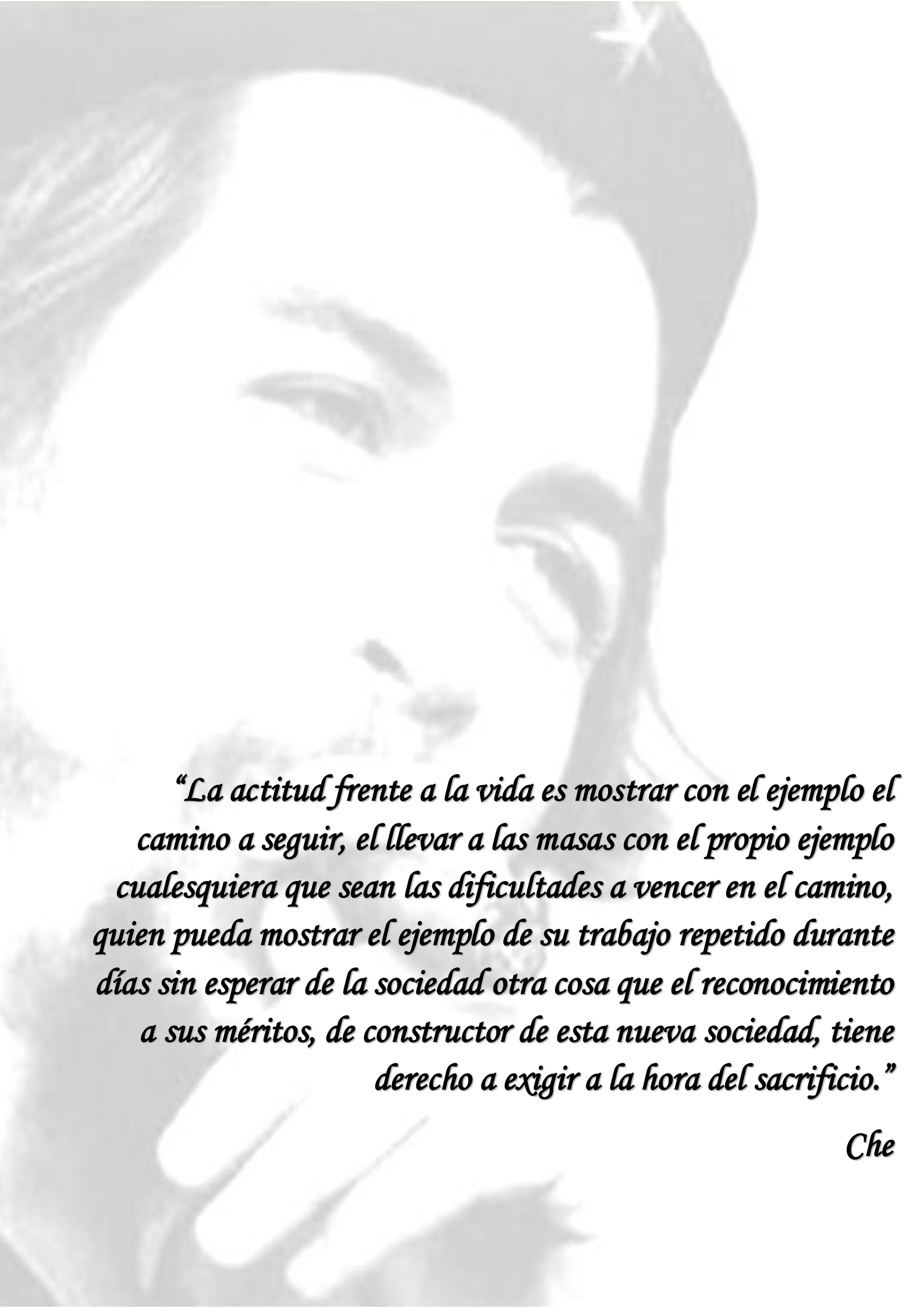
Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yalieski Revé Borges

Ing. Leonardo San Román Labaut

Firma del autor

Firma del tutor



“La actitud frente a la vida es mostrar con el ejemplo el camino a seguir, el llevar a las masas con el propio ejemplo cualesquiera que sean las dificultades a vencer en el camino, quien pueda mostrar el ejemplo de su trabajo repetido durante días sin esperar de la sociedad otra cosa que el reconocimiento a sus méritos, de constructor de esta nueva sociedad, tiene derecho a exigir a la hora del sacrificio.”

Che

AGRADECIMIENTOS

Muchas veces se nos hace difícil describir con palabras lo que el corazón siente, pero también es necesario que muchas de las personas que están a nuestro alrededor tengan una idea de lo que pensamos, es por eso que a través de estas breves palabras quisiera hacerles llegar mi agradecimiento a cada persona que de una forma u otra ha sido parte de mi vida y me ha alentado a seguir adelante y aunque no las pueda mencionar a todas no dejan de ser importantes para mi.

A mis padres principalmente por ser parte de mi vida en todo momento, por brindarme su amor, su apoyo y cariño en todas las horas de mi vida, por estar a mi lado y darme fuerzas para ser cada día mejor para enfrentarme al mundo.

A mi mamita linda a quien no tengo forma de agradecer por todo lo que me ha dado, por ser la madre más bella y buena del mundo, por ser mi sostén, mi guía, sin ti no hubiera sido capaz de convertirme en la persona que soy hoy y tratar de ser mejor, gracias por darme valor, ayudarme a ser fuerte y responsable, por depositar tu confianza en mi, gracias por enseñarme las cosas de la vida y ver el mundo de una manera diferente y por estar junto a mi en todos los momentos, mamita gracias por existir porque sin ti mi vida no tendría sentido.

A mi papito porque a pesar de no estar cada día conmigo me diste todo tu cariño, amor y ternura, por exigirme que fuera una persona preparada y superarme profesionalmente para la vida porque me sirvió para ser lo que hoy soy, gracias a ti me he convertido en alguien con nuevos conocimientos, por darme su apoyo y ayudarme en todo lo que necesito, también por confiar en mi, gracias padre mío por ser alguien importante para mi, y todo lo que me impulsa a ser mejor.

Se que para ustedes esto ha sido muy importante y que sienten un inmenso orgullo al verme graduada porque fue lo que mas han deseado. Aquí esta nuestro triunfo y éxito porque no es solo mío, es de ustedes también. Los Amo con la vida. A mi familia, mis tíos(as) y primos(as) a todos en general porque no alcanzaría la vida para agradecer lo que hacen por mi.

A mis compañeros de grupo por a haber compartido conmigo 5 años de nuevas experiencias, a todos mis amigos y amigas.

En especial a mi eterno amor Jorgito, que me ha hecho pasar los momentos más felices de mi vida y que siempre ha estado conmigo en los buenos y malos momentos. Con su apoyo y su amor me ha hecho sentir la mujer más feliz del mundo.

Gracias a todos por formar parte de mi vida y ser parte de este triunfo.

DEDICATORIA

 *mis padres y mi familia*

RESUMEN

La sociedad de la información, impulsada por un acelerado avance científico y sustentado por el uso generalizado de las potentes y versátiles Tecnologías de la Información y las Comunicaciones (TIC), conlleva cambios que alcanzan todos los ámbitos de la actividad humana. Sus efectos se manifiestan en el terreno educativo donde es cada día mayor. Cuba enfrenta actualmente el desafío de las Tecnologías de la Información y las Comunicaciones, lo que requiere de cambios cualitativos profundos en el sistema educativo cubano para continuar elevando la calidad de la educación para todos. Debido a esto, en un período relativamente joven el Ministerio de Educación (MINED), ha organizado el proceso de producción del software e introducido los mismos en todas las educaciones. Analizando la fuerte línea de desarrollo de productos Software Educativo que se produce en la Universidad de las Ciencias Informáticas (UCI), para el MINED se define el objetivo concreto de este trabajo, que constituye el planteamiento de una propuesta de los elementos arquitectónicos a utilizar en proceso de desarrollo de estos productos. En la investigación se realizaron encuestas y entrevistas, como línea base para desarrollar la propuesta, que daría respuesta al problema en cuestión. Se abordan conceptos que servirán como fundamento teórico de la misma, así como, las características y elementos estructurales más fundamentales del software educativo actual que se produce para el MINED. Como resultado se obtuvo la propuesta de los elementos arquitectónicos, garantizando con dicha propuesta un avance hacia la mejora en la producción y las capacidades del software educativo cubano.

Palabras Claves

Software Educativo, Elementos Arquitectónicos, Arquitectura de Software

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción.....	6
1.2 El software: un nombre nuevo de una práctica ancestral.....	6
1.3 Software educativo con tecnología multimedia y su repercusión en la educación.	7
1.4 Diseño didáctico del software educativo.	9
1.5 Desarrollo centrado en la arquitectura: un acercamiento diferente a la ingeniería de software.....	9
1.5.1 Arquitectura de software	11
1.5.2 Elementos básicos de arquitectura.....	11
1.5.3 Modelo y Visualización Arquitectónico	12
1.5.4 Análisis Arquitectónico	12
1.5.5 El modelo 4+1 en la arquitectura del software	13
1.5.6 Clases de estilos arquitectónicos.....	14
1.5.6.1 Estilos de Flujo de datos	15
1.5.6.2 Estilos centrados en datos	15
1.5.6.3 Estilos de Llamada y Retorno	15
1.5.6.4 Estilo de Código Móvil.....	16
1.5.6.5 Estilos Peer to Peer	16
1.5.7 Patrones arquitectónicos.....	17
1.5.8 Adecuación Patrón Modelo – Vista – Controlador (MVC) en el diseño de SWE....	20
1.5.9 El patrón Modelo – Vista – Controlador – Entidad (MVC-E) como propuesta arquitectónica del software educativo.....	22
1.6 Experiencias anteriores vinculadas al problema.....	22
1.7 Actualidad en la modelación del SWE en Cuba	24
1.8 Situación Problemática	25

1.9 Conclusiones	26
CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN	27
2.1 Introducción.....	27
2.2 Proceso de desarrollo del software educativo	27
2.3 Notaciones en la producción del SWE cubano.....	28
2.4 Características del software educativo en el MINED	29
2.5 Tipologías del Software Educativo	29
2.6 Estructura básica de los programas didácticos	30
2.7 Funcionalidades del Software Educativo	30
2.8 Tipologías de los Programas Didácticos.....	32
2.9. Características del SWE en el MINED	39
2.10 Elementos Estructurales del software educativo MINED.	40
2.11 Descripción de la solución propuesta.....	42
2.11 Conclusiones	46
CAPÍTULO 3: EVALUACIÓN TÉCNICA DE LA PROPUESTA DE SOLUCIÓN	47
3.1 Introducción.....	47
3.2 Guía para la evaluación técnica	47
3.3 Conclusiones	53
CONCLUSIONES GENERALES	54
RECOMENDACIONES	55
REFERENCIAS BIBLIOGRÁFICAS	LVI
BIBLIOGRAFÍA.....	LVIII
ANEXOS.....	LX
GLOSARIO DE TÉRMINOS Y SIGLAS	LXX

INTRODUCCIÓN

Las sucesivas revoluciones científicas-técnicas hasta la actual han ido logrando que la actividad del hombre sea cada vez más racional e intelectual, en la actualidad los seres humanos centran más su accionar en el procesamiento de la información que en la influencia física directa.

El progreso económico y social de un país hoy no solo depende de sus recursos minerales o de la inversión de capital, depende cada vez más de la preparación de su capital humano y sobre la educación descansa la responsabilidad de una adecuada formación y preparación, tanto intelectual como moral, de la mayor parte de la población, sin la que difícilmente se podría avanzar. Ruptura del discurso, tratar de integrar en una idea breve el cambio científico, la educación de las nuevas generaciones y la nueva epistemología. Para alcanzar tal propósito no se puede continuar con los antiguos modelos educativos en los que lo esencial era la transmisión de la información, se impone lograr en los estudiantes *aprender a conocer, aprender a hacer, aprender a ser y aprender a vivir juntos*. (UNESCO, 2001).

Hoy para que el hombre esté a nivel de su tiempo es imprescindible también dominar la tecnología y a la vez aprender con la tecnología. Vivimos en la sociedad de la información, pero que es ante todo *Sociedad de la Formación*. (Vaquero, 1997).

El auge de la informática, surgido en la nueva “era de la información”, permitió que los países del primer mundo pudieran tener un rápido acceso a las tecnologías, mientras que el resto de los países apenas daban los primeros pasos, pues no contaban con las tecnologías y personal calificado necesario para trabajar con ellas.

Es evidente que para los países subdesarrollados, lograr la informatización en todas las esferas de la sociedad, resulta un reto, ya que su problemática fundamental está en lograr la supervivencia de sus pueblos.

Una sociedad informatizada podrá ser más eficaz, eficiente y competitiva, por esta razón Cuba ha identificado desde muy temprano la necesidad y conveniencia de introducir en la práctica social las Tecnologías de la Información y las Comunicaciones. Una de las formas de encaminar a la sociedad hacia el objetivo de un desarrollo sostenible es crear en el hombre nuevo un dominio de la cultura digital, es por ello que el pueblo cubano enfrenta hoy la difícil tarea de informatizar la sociedad.

La Educación Cubana se ha convertido en referencia para muchos países del mundo por el desarrollo y los grandes logros alcanzados durante los 50 años de Revolución, por lo tanto, la informatización del sector educacional es una de las tareas más importantes y complejas a las que se enfrentan hoy los profesionales de la informática en Cuba.

El uso del software educativo (SWE) como medio de enseñanza constituye un elemento de vital importancia para el logro de los objetivos planteados en las transformaciones de la escuela cubana y en un período relativamente joven el Ministerio de Educación (MINED) ha organizado el proceso de producción del software e introducido los mismos en todas las educaciones.

El software educativo cubano, como respuesta y representante del sistema educacional de la isla, posee un conjunto de características que lo diferencian sustancialmente del que se produce en el resto del mundo; potenciando áreas como: (1) Procesamiento de volúmenes de información considerables en varios formatos de presentación (*animaciones, videos, imágenes, sonidos, textos e hipertextos*). (2) Utilización de técnicas de Inteligencia Artificial (IA) para el tratamiento de entornos educacionales a la medida del usuario. (3) Concepción de seguimientos pedagógicos o desarrollo de la traza de utilización y avance cognitivo del usuario, lo que implica la utilización de bases de datos. (4) Posibilidad de retroalimentación del tutor o profesor guía del proceso docente educativo donde se utiliza el software. (5) Posibilidad del profesor de adecuar el funcionamiento de la aplicación a las características de cada uno de los estudiantes o grupos de estos. (6) Posibilidad del profesor de adicionar elementos multimedia al software para un mejoramiento y actualización del mismo.

A raíz de lo antes expuesto surge el siguiente **problema científico**: Insuficiente conocimiento de los elementos estructurales y lógicos comunes del software educativo para el MINED.

El modelo descrito genera como **objeto de estudio** de esta investigación: El proceso de concepción y diseño pedagógico del software educativo, enmarcado en el proceso de concepción y diseño pedagógico del software educativo para el MINED como **campo de acción**.

Para dar solución a nuestro problema científico se planteó como **objetivo general**: Elaborar una propuesta de los elementos arquitectónicos a utilizar en la producción de software educativo para el MINED.

Al mismo tiempo se plantean dos **objetivos específicos** para un mejor logro del objetivo general descrito anteriormente:

- Definir los componentes conceptuales comunes, significativos y con capacidad de generalización de los elementos estructurales y lógicos del software educativo producido para el MINED.
- Definir los elementos arquitectónicos necesarios en la producción del software educativo para el MINED.

Para guiar la investigación se planteó la siguiente **idea a defender**:

Con la aplicación de los elementos arquitectónicos propuestos para producción del software educativo para el MINED, se garantizaría un mejor diseño funcional y pedagógico, la disminución del tiempo de desarrollo de este tipo de aplicaciones, así como, un aumento en la satisfacción de las necesidades de los mismos y paralelo a esto incrementaría la ventaja competitiva de la organización frente a sus competidores.

La investigación científica se guió por el conjunto de **tareas** que se describen a continuación para dar cumplimiento al objetivo tanto general como específico planteado:

- Estudiar el estado del arte del tema en Cuba y el mundo.
- Estudiar la documentación generada a partir del diseño y concepción pedagógica del software educativos para el MINED.
- Estudiar las condiciones actuales se desarrolla el software educativo en Cuba, y en particular el MINED.
- Identificar los elementos estructurales comunes y significativos como resultado de los diferentes procesos de diseño y concepción pedagógica utilizados para el MINED.
- Identificar los elementos funcionales comunes y significativos como resultado de los diferentes procesos de diseño y concepción pedagógica utilizados en el mundo.
- Proponer los elementos arquitectónicos y logros a utilizar en la producción del software educativo para el MINED.
- Validar según el método de expertos los elementos propuestos.

Para el desarrollo de la investigación se trabajó con la población conformada por todos los productos de software educativo que se han desarrollado en la UCI, para la institución del MINED. A dicha población se le realizó la técnica de muestreo aleatorio simple, donde se hizo un listado de todos los productos de software educativo existentes en la UCI, asignándole un número a cada uno y utilizando un segmento de la tabla de números aleatorios de (Spiegel M. R 1977) para la selección de la muestra, trabajando con las **unidades de estudio** sobre los productos de software educativo.

En el desarrollo de la investigación científica se han utilizado un conjunto de **métodos de investigación científicos** para la obtención, procesamientos y llegada a conclusiones. De los que podemos citar los siguientes:

Métodos teóricos:

Histórico _ Lógico y Dialéctico: para el estudio crítico de trabajos anteriores y tomar estos como punto de referencias y comparación de los resultados alcanzados.

Analítico _ Sintético: al descomponer el problema de investigación en elementos por separados y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta.

Métodos empíricos:

Entrevista: utilizado para obtener información verbal y real de los distintos miembros internos y externos a la actividad de producción en los proyectos para el MINED, así como a los especialistas pedagógicos inmersos en la investigación.

Encuesta: a líderes y estudiantes de los proyectos de software educativo para el MINED, para recopilar la información necesaria que permita validar la solución propuesta.

Métodos cualitativos:

Teoría Fundamentada: Para la obtención de la información necesaria a través de cualquier fuentes de datos (entrevistas, observaciones de campo, documentos de todo tipo).

Con la aplicación de la estrategia: *Método de la comparación constante* (El investigador codifica y analiza los datos de forma simultánea para desarrollar conceptos.)

El trabajo que se presenta consta de 3 capítulos. **El primero**, “Fundamentación Teórica”, abarca los principales conceptos y es la base de este trabajo. **El segundo**, está enmarcado en la solución propuesta de la investigación. **El tercero** “Evaluación Técnica de la Propuesta de solución”, donde se determina la probabilidad de éxito que tiene la propuesta. Para ello se utiliza el método de experto, que permite tomar decisiones para aceptar o no la propuesta de acuerdo con diferentes criterios de evaluación. Para finalizar se presentan las conclusiones generales, las recomendaciones, las referencias bibliográficas, la bibliografía, un glosario de términos utilizados y el conjunto de anexos para un mejor entendimiento de lo expuesto a lo largo de este trabajo.

1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

“La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.”(Kruchten, Philippe)

La frase anterior, expresada por Kruchten Philippe, una de las personalidades más reconocidas en el campo de la Ingeniería de Software (ISW), da pie al contenido del presente capítulo, expresiones como arquitectura, patrones arquitectónicos, estilos arquitectónicos, modelado y otros son la base del mismo, con el objetivo principal de situar al lector en el contenido de la investigación realizada.

1.2 El software: un nombre nuevo de una práctica ancestral.

Software, hardware y computación, son términos que se asocian con la modernidad cuando en realidad son nombres nuevos de prácticas ancestrales. El software no surge con los equipos electrónicos, -aunque es con ellos que adopta el nombre- está presente desde el empleo de ábacos o sumadoras mecánicas. Sin embargo, en estos casos, el software no se encuentra incorporado en el equipo. Es aportado por el operario. Un ejemplo de esto fue la máquina analítica de Charles Babbage, que incidentalmente, tuvo su software, y fue una amiga de éste, la legendaria lady Lovelace, quien aportó el software que no se llegó a usar, dado que la máquina nunca se completó.

Se denomina software, a todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Dentro de las tipologías de software podemos encontrar los Software de sistema, que constituyen la parte que permite funcionar al hardware y se encargan de aislar tanto como sea posible al programador de aplicaciones de los detalles del computador particular que se use, especialmente de las características físicas de la memoria, dispositivos de comunicaciones, impresoras, pantallas, teclados, etcétera. Los Software de programación, proporcionan herramientas para ayudar al programador a escribir programas informáticos y a usar diferentes lenguajes de programación de forma práctica y por ultimo los Software de aplicación, que permiten a los usuarios llevar a cabo una o varias tareas más específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Dentro de esta última clasificación se encuentra el software educativo que no son mas que los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar el proceso de enseñanza (tutoriales, publicitarios), el proceso de aprendizaje (simuladores, entrenadores), o el proceso de enseñanza-aprendizaje (mixtos). Para la producción de software educativo, en los últimos tiempos se ha incrementado mucho la utilización de la tecnología multimedia, ya que brinda muchas ventajas para lograr los objetivos perseguidos en un software educativo.

1.3 Software educativo con tecnología multimedia y su repercusión en la educación.

En la dinámica del mundo actual, se está viviendo un avance tecnológico que necesita nuevas concepciones de la educación del individuo, éste requiere de alfabetización tecnológica para poder actuar con pertinencia en entornos cada vez más variados de combinación e integración más mediática. Dentro del ambiente más mediático, se destaca la tecnología multimedia por su versatilidad y múltiples usos. Las bondades de la naturaleza de la tecnología multimedia han provocado su aplicación en cada vez mayor cantidad de áreas del quehacer humano, a lo que no ha escapado la educación, para la cual se considera que puede resultar una herramienta muy útil en el proceso de enseñanza y de aprendizaje, al permitir que el alumno sienta que su escuela es una prolongación de su manera cotidiana de comunicarse y no como paradójicamente se ha venido señalando hasta ahora, que la escuela es precisamente el lugar donde el alumno tiene menos posibilidades de expresarse.

Los productos educativos que utilizan la tecnología multimedia son instrumentos muy poderosos para una enseñanza activa, basada en el descubrimiento, la interacción y la experimentación. Su aporte principal reside en su contribución a la realización de una pedagogía activa además de que está demostrado que los recursos multimediales son

sumamente atractivos y pueden ayudar a generar la ilusión de motivar al alumno y producir mejores aprendizajes.

Muchos autores coinciden en que los sistemas con tecnología multimedia ofrecen aspectos positivos y negativos que conviene tener presentes para potenciar unos y minimizar otros.

Dentro de estos aspectos positivos se pueden mencionar algunos como:

- Ofrecen la posibilidad de controlar el flujo de información.
- Un programa multimedia bien diseñado no corre el peligro de obsolescencia, puesto que pueden actualizarse con facilidad los contenidos con pequeños cambios en el software.
- Puede darse una mejora en el aprendizaje ya que el alumno avanza por el sistema según su ritmo individual de aprendizaje. Puede pedir información, animarse a penetrar en temas nuevos cuando tenga dominado los anteriores, seguir sus intereses personales.
- Puede incrementarse la retención. La memorización de núcleos de información importantes aumentará significativamente gracias a la interacción y a la combinación de imágenes, gráficos, textos,... junto a las simulaciones con representaciones de la vida real.
- Puede aumentar la motivación y el gusto por aprender. El aprendizaje se convierte de este modo en un proceso lúdico.
- Puede lograrse una mayor consistencia pedagógica, ya que la información contenida es la misma en distintos momentos y para diferentes alumnos.
- La metodología de trabajo, dentro de su variedad, es homogénea.
- Puede convertirse en forma creciente y en función de la evolución de las tecnologías que lo sustentan en uno de los medios de instrucción de más calidad.

Sin embargo, la experiencia está mostrando también que, mal elegidos en función del grupo escolar e inadecuadamente utilizados:

- Potencian la fragmentación del conocimiento.
- Producen saturación de información, elevan los umbrales de impacto y velocidad en las imágenes que un alumno requiere como estímulo para interesarse.
- Fomentan la pasividad frente a la pantalla.
- En la medida en que utilizan atajos visuales para la comprensión desalientan los procesos más abstractos de inferencia.
- Centran la atención en aspectos superficiales y no relevantes del conocimiento

1.4 Diseño didáctico del software educativo.

El diseño de programas educativos, cuando responde a una planificación estricta y cuidadosa desde el punto de vista didáctico, puede no verse correspondido en la puesta en práctica, dándose una utilización totalmente casual [...], sin embargo, también puede ocurrir la situación inversa: un determinado software no diseñado específicamente, con unas metas difusas y sin unos determinados objetivos puede ser utilizado con una clara intencionalidad.
(Urbina, 1999)

La anterior cita es una realidad, pero un trabajo serio no puede apostar a la casualidad, se deben encaminar los esfuerzos para lograr el diseño y elaboración del software con una intencionalidad educativa explícita, proponiendo y/o facilitando determinadas estrategias de aprendizaje que se vean correspondidas por un uso consecuente, creativo y enriquecedor, en la práctica.

En el diseño del material educativo se debe tener en cuenta que uno u otro presupuesto teórico del aprendizaje incide; manifestándose en la selección y organización de los contenidos, en la forma de adaptación a los usuarios, en las estrategias de enseñanza de los mismos, en el grado de control sobre las actividades de los estudiantes, en el diseño de las pantallas y en la forma que el usuario se comunica con el programa. Pero independiente de la finalidad proyectada en el diseño del software, la concepción del educador de cómo se ha de utilizar será determinante siempre.

1.5 Desarrollo centrado en la arquitectura: un acercamiento diferente a la ingeniería de software.

Todo estudiante de ciencias computacionales ha practicado el desarrollo de software en cursos de introducción generalmente enfocados en programación y algoritmos. Estos cursos típicamente introducen sintaxis básicas de los lenguajes de computación como las decisiones y ciclos junto con principios básicos de diseño como la abstracción, modularidad y encapsulamiento. Para la mayoría de los problemas pequeños estos principios son suficientes. Sin embargo, conforme la complejidad crece, estas técnicas son insuficientes para asegurar un producto exitoso que permita que los requerimientos y las expectativas de calidad sean alcanzados. Los grandes proyectos de software tienen grandes posibilidades de falla: el reporte del grupo Standish, declara que “cerca de un tercio de los proyectos son cancelados antes de completarse y más de la mitad sufren serios incrementos en su costo.”

La ingeniería de software a gran escala es una actividad inherentemente compleja que involucra la creación y manipulación, por varias personas, de un gran número de artefactos

frecuentemente intangibles y muy dinámicos. Éstos pueden incluir especificaciones de requerimientos, diseños de software de alto nivel, código fuente, información de pruebas, y mantenimiento y necesidades de evolución posteriores a la implementación. Debido a la complejidad de la actividad, muchos de los esfuerzos se han dedicado a mejorar el proceso de ingeniería de software, esto es, los pasos a seguir durante la construcción del software. Estándares ampliamente aceptados para alcanzar la calidad, como el conocido como *Capability Maturity Model* (CMM) [Modelo de Capacidad de Madurez] se enfocan exclusivamente en el proceso. Aun a pesar de estas mejoras al proceso, no necesariamente veremos un mejoramiento significativo correspondiente en el software que construyamos. *“Un buen proceso no garantiza un buen producto.”*

Existe otro camino para mejorar la calidad del software y las probabilidades de éxito de los proyectos. La arquitectura de software es una disciplina capaz de conectar e integrar a personas, actividades y productos involucrados en la ingeniería de software. La arquitectura de software permite a los ingenieros mayor control e introspección en el sistema en el proceso de desarrollo desde sus inicios, y promueve la temprana identificación y prevención de problemas. Como resultado, la arquitectura puede guiar el proyecto al éxito en vez de librarlo al fracaso por falta de entendimiento.

Todo sistema de software, desde el más pequeño hasta los sistemas multi-organizacionales tienen una arquitectura que puede variar en calidad. La arquitectura captura el conjunto de decisiones principales de diseño que se hacen sobre un sistema. Las decisiones de diseño son elecciones hechas pensando en cómo el sistema se desarrollará y cómo funcionará, y estas elecciones pueden incluir estructura, organización, funcionalidad, comportamiento, o más propiedades no funcionales como la usabilidad y estética [de la interfaz de usuario]. La importancia de estas decisiones de diseño varía para cada sistema de software y está en función de los interesados en el sistema, sus preocupaciones, y sus necesidades específicas. Una cuestión clave es que, a pesar de que la arquitectura es fundamentalmente una actividad centrada en el diseño, afecta a todo el ciclo de vida. Las decisiones de la arquitectura se concentran en lo que es esencial en un sistema la influencia de los requerimientos y las actividades importantes con el diseño colaborativo, el desarrollo e implementación del sistema, la planeación de su evolución y su adaptación.

Adquirir todos los beneficios de la arquitectura de software requiere una aplicación apropiada de la disciplina. Los sistemas con buena arquitectura son propensos al éxito; mientras sistemas con arquitecturas pobres están condenados a fallar.

1.5.1 Arquitectura de software

La Arquitectura de software (en lo adelante AS) tiene sus raíces en 1968, donde Edsger Dijkstra, de la Universidad Tecnológica de Eindhoven en Holanda y Premio Turing 1972, propuso que se estableciera una estructuración correcta de los sistemas de software antes de lanzarse a programar, escribiendo código de cualquier manera . Dijkstra, quien sostenía que las ciencias de la computación eran una rama aplicada de las matemáticas y sugería seguir pasos formales para descomponer problemas mayores, fue uno de los introductores de la noción de sistemas operativos organizados en capas que se comunican sólo con las capas adyacentes

En el año 2000 la IEEE hace la definición “oficial” de AS en su documento IEEE 1471, que reza así:

“La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

En la actualidad se define a la arquitectura de software como:

- Vista estructural de alto nivel.
- Define estilo o combinación de estilos para una solución.
- Se concentra en requerimientos no funcionales.
- Los requerimientos funcionales se satisfacen mediante modelado y diseño de aplicación.
- Esencial para éxito o fracaso de un proyecto.

1.5.2 Elementos básicos de arquitectura

Las buenas arquitecturas, en el nivel más básico, capturan la estructura de un sistema de software en términos de elementos arquitectónicos de alto nivel. Estos elementos son **componentes** y **conectores**, ligados entre ellos en **configuraciones** específicas. Los componentes son elementos arquitectónicos que son, primordialmente, responsables de la computación. Conforme los componentes involucran los aspectos funcionales del sistema, las estrategias tradicionales de diseño, como la descomposición funcional, pueden contribuir significativamente a la identificación y restricciones de componentes. Los conectores son elementos que solo son responsables de facilitar y administrar la comunicación entre los componentes. Los conectores permiten que los componentes independientes interactúen sin la necesidad de un conocimiento directo mutuo. Al ser la independencia de componentes la

llave de la reutilización, los conectores pueden significar un ahorro significativo de costo. Finalmente, las configuraciones combinan componentes y conectores en una unidad, única y coherente. Las configuraciones dirigen el comportamiento del sistema de software.

Estos componentes y conectores arquitectónicos no son abstracciones lógicas únicamente útiles durante el diseño, sino que deben ser explícitamente modelados, conservados e implementados para que acompañen al sistema y sean utilizados durante su desarrollo y despliegue.

1.5.3 Modelo y Visualización Arquitectónico

El modelado arquitectónico es el proceso de capturar y documentar decisiones del diseño arquitectónico y puede hacerse de diferentes maneras. Algunos arquitectos y desarrolladores capturan las decisiones de diseño arquitectónico usando documentos de lenguaje natural y diagramas abstractos de "cajas y flechas". Mientras estas formas informales de documentar son útiles en muchos aspectos, generalmente carecen del rigor y la precisión necesarios para la descripción comprensible de la arquitectura.

La clave para habilitar la arquitectura en otras áreas, además del diseño, es lograr un nivel apropiado de rigor y precisión en el modelado arquitectónico. Los modelos buenos proveen la base para el análisis arquitectónico, la comprensión y la comunicación del diseño del sistema, y guiar la evolución del mismo.

La selección de la notación de modelo y las técnicas de visualización también deben de ser en función del problema. Desde el lenguaje natural pasando por UML hasta los ADL, los arquitectos tienen una gran variedad de notaciones y visualizaciones para escoger. La selección también debe ser en función de los problemas a solucionar previamente identificados. Los buenos arquitectos deben de estar bien informados de las armas disponibles en notación de modelado, y hacer uso de ellas libremente -un error común en la arquitectura de modelos es limitar el modelado a una simple notación por conveniencia, por adopción, o por minimizar esfuerzo.

1.5.4 Análisis Arquitectónico

Al análisis arquitectónico le interesa contestar preguntas que no puedan ser respondidas con el simple conocimiento de la arquitectura de un sistema. Por ejemplo, "¿El sistema será confiable?". Existen muchos métodos de análisis arquitectónico que pueden ser aplicados en la arquitectura cubriendo los que se hacen de forma manual hasta los que son hechos totalmente automáticos. El objetivo final del análisis arquitectónico es, obviamente, el

análisis totalmente automático: la arquitectura es definida, y una herramienta determina si la arquitectura alcanza cierta calidad o tiene cierta propiedad. Esto será posible en el grado de que dependa directamente de cómo la arquitectura será especificada.

La habilidad de un análisis en particular para revisar la arquitectura de software contra diferentes problemas a resolver debe ser tomada en cuenta cuando se escoge la notación de modelado y profundidad de detalle.

1.5.5 El modelo 4+1 en la arquitectura del software

Perry y Wolf describen una arquitectura software como: “Arquitectura Software= {Elementos, Formas, Fundamento/Restricciones}”

Es muy complejo capturar la arquitectura software en un sólo modelo (o diagrama). Para manejar esta complejidad se representan diferentes aspectos y características de la arquitectura en múltiples vistas. Una vista es “una presentación de un modelo, la cual es una descripción completa de un sistema desde una particular perspectiva” (Kruchten, 1995).

“El modelo más aceptado a la hora de establecer las vistas necesarias para describir una arquitectura software es el modelo 4+1” (Kruchten, 1995). (Ver figura1)

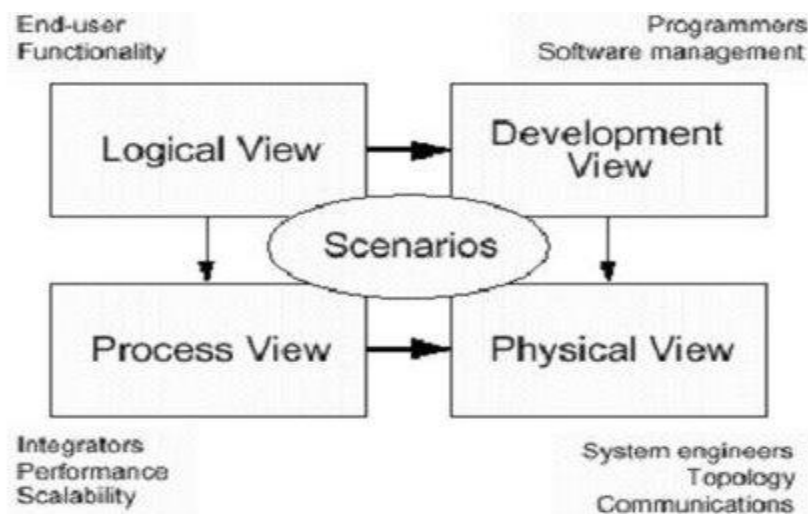


Figura 1 Vistas Arquitectónicas

Este modelo define 4 vistas principales:

- Vista lógica
- Vista de procesos
- Vista de desarrollo
- Vista física

Y una vista, que se muestra y traza en cada una de las anteriores y que esta formada por las necesidades funcionales que cubre el sistema y que en ocasiones identificamos como **vista de Caso de Uso**. De donde deducimos que según este modelo, la arquitectura es en realidad evolucionada desde escenarios.

Cada vista es descrita usando su particular y adecuada notación (por ejemplo existen diagrama UML, que se adaptan mas a una vista que otra). Para cada vista los arquitectos puedes escoger cierto estilo arquitectónico (patrón arquitectónico) permitiendo la coexistencia de varios patrones en el sistema.

Por ultimo podemos comentar que el modelo de vista 4+1 es también generico: otra notaciones y herramientas a parte de UML, pueden usarse y cualquier método de diseño, especialmente para las descomposiciones lógicas y de proceso.

1.5.6 Clases de estilos arquitectónicos

En el texto fundacional de la AS, Perry y Wolf establecen el razonamiento sobre estilos de arquitectura como uno de los aspectos fundamentales de la disciplina. Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software, mientras que las formas complejas se articulan mediante composición de los estilos fundamentales.

Concisamente descriptos, los estilos conjugan elementos (o “componentes”, como se los llama aquí), conectores, configuraciones y restricciones. Al estipular los conectores como elemento de juicio de primera clase, el concepto de estilo, incidentalmente, se sitúa en un orden de discurso y de método que el modelado orientado a objetos en general y UML en particular no cubren satisfactoriamente. La descripción de un estilo se puede formular en lenguaje natural o en diagramas, pero lo mejor es hacerlo en un lenguaje de descripción arquitectónica o en lenguajes formales de especificación.

A diferencia de los patrones de diseños, que son centenares, los estilos se ordenan en seis o siete clases fundamentales y unos veinte ejemplares, como máximo. Es digno de señalarse el empeño por subsumir todas las formas existentes de aplicaciones en un conjunto de dimensiones tan modestas. Las arquitecturas complejas o compuestas resultan del agregado o la composición de estilos más básicos.

Algunos de los principales estilos arquitectónicos se usan en la actualidad están divididos por Clases de Estilos las que engloban una serie de estilos arquitectónicos específicos.

1.5.6.1 Estilos de Flujo de datos

Esta familia de estilos enfatiza la reutilización y la modificabilidad. Es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos.

Tuberías y Filtros (estilo arquitectónico específico): Una tubería es una popular arquitectura que conecta componentes computacionales a través de conectores, de modo que el procesamiento de datos se ejecuta como un flujo. Los datos se transportan a través de las tuberías entre los filtros, transformando gradualmente las entradas en salidas. Este estilo se percibe como una serie de transformaciones sobre sucesivas piezas de los datos de entrada.

1.5.6.2 Estilos centrados en datos

Esta familia de estilos enfatiza la integrabilidad de los datos. Se estima apropiada para sistemas que se fundan en acceso y actualización de datos en estructuras de almacenamiento.

Arquitecturas de Pizarra o repositorio: Esta arquitectura está compuesta por dos componentes principales: una estructura de datos que representa el estado actual y una colección de componentes independientes que operan sobre él. Estos sistemas se han usado en aplicaciones que requieren complejas interpretaciones de proceso de señales (reconocimiento de patrones, reconocimiento de habla).

1.5.6.3 Estilos de Llamada y Retorno

Esta familia de estilos enfatiza la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas en gran escala.

Modelo -Vista-Controlador (MVC): en este estilo arquitectónico se divide la aplicación en tres capas. Una de ellas (la vista) será la encargada de gestionar la interfaz y todo lo relacionado con la interacción entre la aplicación y el usuario, otra (el modelo) será la encargada de hacer el trabajo real de la aplicación, y la última (el controlador) sirve para desacoplar ambas, de forma que, si fuera necesario, una vista pueda ser actualizada por varios modelos, o un modelo pueda proveer de datos varias vistas.

Arquitectura en Capas: En este estilo arquitectónico cada capa proporciona servicios a la capa superior y se sirve de las prestaciones que le brinda la inferior, al dividir un sistema en capas, cada capa puede tratarse de forma independiente, sin tener que conocer los detalles de las demás. La división de un sistema en capas facilita el diseño modular, en la que cada capa encapsula un aspecto concreto del sistema y permite además la construcción de

sistemas débilmente acoplados, lo que significa que si se minimiza las dependencias entre capas, resulta más fácil sustituir la implementación de una capa sin afectar al resto del sistema.

Arquitectura Orientada a Objetos: Los componentes de este estilo son los objetos, o más bien instancias de los tipos de dato abstractos. En la caracterización clásica de David Garlan y Mary Shaw, los objetos representan una clase de componentes que ellos llaman manager, debido a que son responsables de preservar la integridad de su propia representación. Un rasgo importante de este aspecto es que la representación interna de un objeto no es accesible desde otros objetos.

Arquitectura basada en Componentes: Los sistemas de software basados en componentes se basan en principios definidos por una ingeniería de software específica. Los componentes son las unidades de modelado, diseño e implementación, las interfaces están separadas de las implementaciones, y conjuntamente con sus interacciones son el centro de incumbencias en el diseño arquitectónico. Los componentes soportan algún régimen de introspección, de modo que su funcionalidad y propiedades puedan ser descubiertas y utilizadas en tiempo de ejecución.

1.5.6.4 Estilo de Código Móvil

Esta familia de estilos enfatiza la portabilidad. Ejemplos de la misma son los intérpretes, los sistemas basados en reglas y los procesadores de lenguaje de comando.

Arquitectura de Máquinas Virtuales: Esta arquitectura se conoce como intérpretes basados en tablas ó sistemas basados en reglas. Estos sistemas se representan mediante un pseudo-programa a interpretar y una máquina de interpretación. Estas variedades incluyen un extenso espectro que está comprendido desde los llamados lenguajes de alto nivel hasta los paradigmas declarativos no secuenciales de programación.

1.5.6.5 Estilos Peer to Peer

Esta familia se conoce también como componentes independientes, enfatiza la modificabilidad por medio de la separación de las diversas partes que intervienen en la computación. Consiste por lo general en procesos independientes o entidades que se comunican a través de mensajes.

Arquitecturas Basadas en Eventos: Estas se han llamado también arquitectura de invocación implícita, estas se vinculan con sistemas basados publicación-suscripción. La

idea dominante en la invocación implícita es que, en lugar de invocar un procedimiento en forma directa un componente puede anunciar mediante difusión uno o más eventos.

Arquitecturas Orientadas a Servicios (SOA en inglés): Esta construye toda la topología de la aplicación como una topología de interfaces, implementaciones y llamados a interfaces; es una relación entre servicios y consumidores de servicios, ambos lo suficientemente amplios como para representar una función de negocio completa.

Arquitecturas Basadas en Recursos: Esta define recursos identificables y métodos para acceder y manipular el estado de esos recursos. El caso de referencia es nada menos que la World Wide Web, donde los URIs identifican los recursos y HTTP es el protocolo de acceso. El argumento central es que HTTP mismo, con su conjunto mínimo de métodos y su semántica simplísima, es suficientemente general para modelar cualquier dominio de aplicación.

1.5.7 Patrones arquitectónicos

Los patrones arquitectónicos expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Los patrones de arquitectura representan el nivel más alto en el sistema.

Al contrario de los estilos arquitectónicos los patrones son muchos y a su vez muy variados y es casi imposible revisar todos los patrones que existen a la hora de hacer una determinada aplicación, por eso se recomienda el uso de los patrones que estén asociados en cada uno de los estilos que se seleccionen para el desarrollo de la arquitectura.

Entre los patrones de arquitectura más utilizados actualmente, podemos citar ***el patrón en capas*** y ***el patrón Modelo-Vista-Controlador***, los mismos serán referenciados a continuación:

Patrón en Capas

Donde la separación de responsabilidades es puramente lógica y podemos encontrar una capa de presentación con las responsabilidades de recibir información del usuario, y pasarla a las capas inferiores, así como recibir información de estas capas y formatearla para presentársela al usuario. Otra capa mas abajo llamada generalmente de lógica de negocio que es donde se implementan todas las reglas de negocio definidas para la aplicación, y finalmente una última capa, mal llamada de datos, que se debe llamar de acceso a datos

que alberga la responsabilidad de acceder a los repositorios de datos y de ser posible abstraer a las capas superiores de las interioridades de estos repositorios (Ver figura 2)

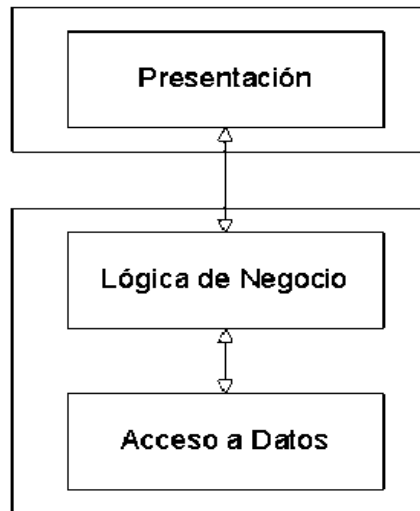


Figura 2 Ejemplo de una arquitectura de tres capas.

Capa de presentación: es la que ve el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

Capa de datos: es donde residen los datos y es la encargada de acceder a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Patrón Modelo-Vista-Controlador

El (Modelo-Vista-Controlador, en adelante MVC) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. Según uno de los *Sistemas de Patrones de Arquitectura* más extendido en el mundo: *Pattern Oriented Software Architecture*, publicado por Buschmann en 1996, el patrón que se analiza en este trabajo (MVC), se sitúa en la

tercera de las cuatro categorías en las cuales clasifica a los patrones: *Del barro a la estructura* (From mud to Structure), *Sistemas Distribuidos* (Distributed Systems), *Sistemas Interactivos* (Interactive Systems) y *Sistemas Adaptables* (Adaptable Systems). De igual forma si utilizamos otro de los sistemas más difundidos: *Pattern of Enterprise Application Architecture*, descrito recientemente por Fowler en el pasado 2003, lo ubica en la tercera de las siete categorías que se mencionan a continuación: *Patrones de Lógica del Dominio* (Domain Logic Patterns), *Patrones de Mapeo a Bases de Datos Relacionales* (Mapping to Relational Database Patterns), *Patrones de Presentación Web* (Web Presentation Patterns), *Patrones de Distribución* (Distribution Patterns), *Patrones de Concurrencia Offline* (Offline Concurrency Patterns), *Patrones de Estado de Sesión* (Session State Patterns) y *Patrones Base* (Base Patterns).

El modelo divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. (Ver Figura 3):

Descripción del patrón:

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

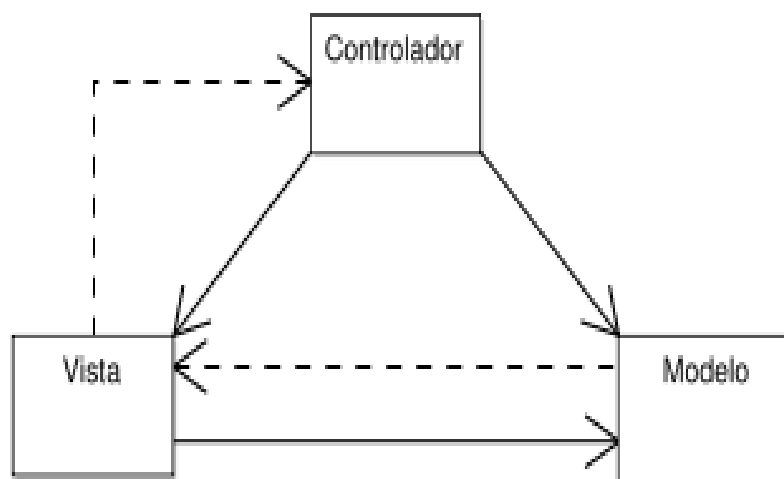


Figura 3 Ejemplo del patrón Modelo-Vista-Controlador

1.5.8 Adecuación Patrón Modelo – Vista – Controlador (MVC) en el diseño de SWE

Hoy día en cualquier lugar del mundo los que construyen aplicaciones informáticas centran su atención en dos aspectos fundamentales: (1) ¿cómo lograr construir mejores aplicaciones en menos tiempo? y (2) ¿cómo utilizar mayor cantidad de estándares en el diseño de las aplicaciones que soporte la primera interrogante planteada y permita mayor reutilización del código y mejores mantenimientos de los sistemas desarrollados? El software educativo no está ajeno a esta problemática, incluso por su concepción, complejidad y variabilidad en el tiempo, se pudiera decir que son el tipo de aplicaciones informáticas que mayor necesidad tienen de darle respuesta a ambas preguntas planteadas.

“La calidad de diseño de la interacción de los objetos y la asignación de responsabilidades presentan gran variación. Las decisiones poco acertadas dan origen a sistemas y componentes frágiles y difíciles de mantener y entender, reutilizar o extender. Una implementación hábil se funda en los principios cardinales que rigen un buen diseño orientado a objetos.” (Larman 1999)

La administración o gestión de datos engloba dos áreas distintas de interés: (1) la administración (gestión) de datos críticos para la propia aplicación, y (2) la creación de infraestructura para el almacenamiento y recuperación de los objetos. En general, la administración de datos se diseña en forma de capas o en paquetes o subsistemas. La idea es aislar de forma lógica desde el análisis y luego en el diseño, los requisitos de bajo nivel que manipulan las estructuras de datos, de los requisitos de alto nivel para manejar los atributos del sistema en desarrollo. (Pressman, 2002)

Por las razones expuestas, los diseñadores cubanos han planteado un conjunto de modificaciones sobre la base del patrón MVC y lo estandarizado para el análisis y el diseño por el Lenguaje Unificado de Modelado (UML) que permita acercar a dicho patrón a las necesidades para este tipo de sistemas informáticos.

Se han desarrollado a lo largo de los años, desde la presentación de este patrón a la comunidad científica 3 variantes fundamentales, que se presentan brevemente a continuación.

Variante I: (Figura 4)

Variante en la cual no existe ninguna comunicación entre el Modelo y la Vista y esta última recibe los datos a mostrar a través del Controlador.

Variante II: (Figura 5)

Variante en la cual se desarrolla una comunicación entre el Modelo y la Vista, donde esta última al mostrar los datos los busca directamente en el Modelo, dada una indicación del Controlador, disminuyendo el conjunto de responsabilidades de este último.

Variante III: (Figura 6)

Variante en la cual se diversifica las funcionalidades del Modelo teniendo en cuenta las características de las aplicaciones multimedia, donde tienen un gran peso las medias utilizadas en estas.

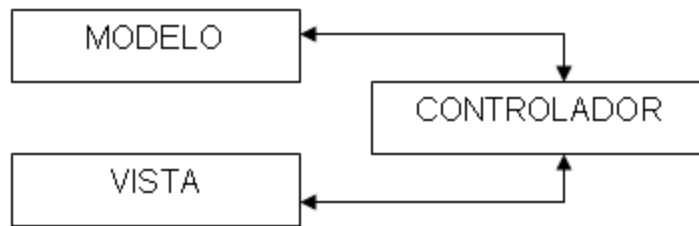


Figura 4 Variante inicial del Patrón MVC

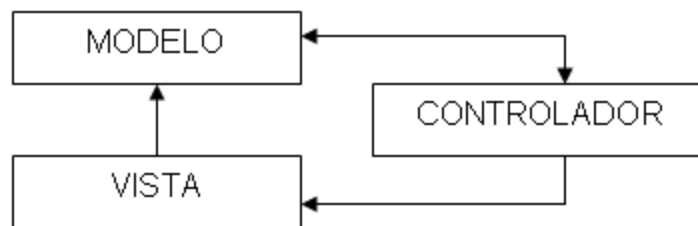


Figura 5 Variante Intermedia del Patrón MVC

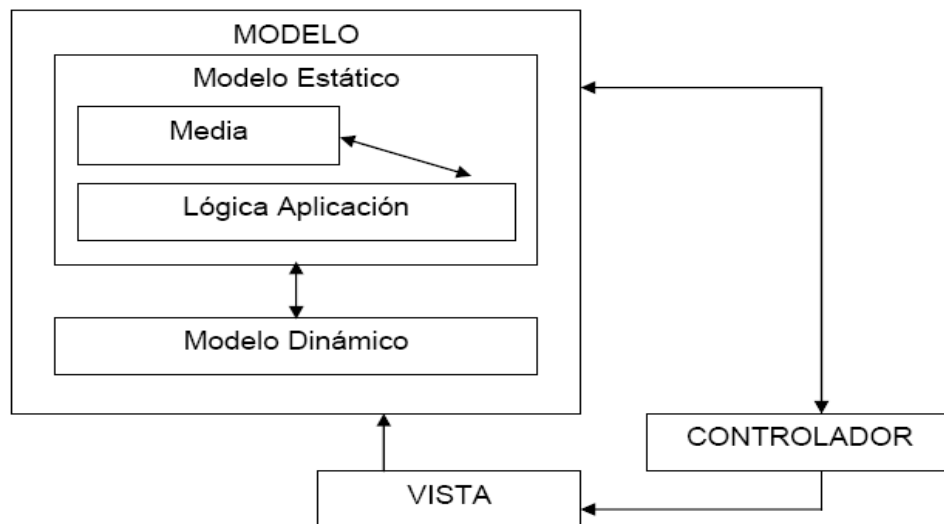


Figura 6 Variante modificada para Aplicaciones Multimedia del MVC, conocida como MVC_{MM}

1.5.9 El patrón Modelo – Vista – Controlador – Entidad (MVC_E) como propuesta arquitectónica del software educativo.

Para poder comprender lo que es el Software (y consecuentemente la ingeniería del Software), es importante examinar las características del Software que lo diferencian de otras cosas que los hombres pueden construir. El Software es un elemento del sistema que es lógico, en lugar de físico. Por tanto el software tiene unas características considerablemente distintas a la del hardware. (Pressman, 2002)

Durante las dos pasadas décadas, se han desarrollado un gran número de métodos de modelado. Los investigadores han identificado los problemas del análisis y sus causas y han desarrollado varias notaciones de modelado y sus correspondientes conjuntos de heurísticas para solucionarlos (...) Se emplean modelos para poder comunicar de forma compacta las características de la función y su comportamiento. Se aplica la partición para reducir la complejidad. Son necesarias las visiones esenciales y de implementación del software para acomodar las restricciones lógicas impuestas por los requisitos del procesamiento y las restricciones físicas impuestas por otros elementos del sistema. (Pressman, 2002)

Este enfoque unido a las características distintivas del software educativo producido en Cuba y descritas en la introducción de este trabajo, ha traído como consecuencia el análisis de una variante de solución al **MVCM** para las aplicaciones educativas cubanas, donde se encarguen las responsabilidades de la clase modelo concernientes al procesamiento y almacenamiento de la información persistente de las aplicaciones, incorporando una nueva clase al modelo denominada **Modelo-Entidad**, con dos tipos fundamentales, la clase **Modelo-Entidad-Media** y **Modelo-Entidad-Persistente**. La primera de estas con la responsabilidad de agrupar las clases que identifican las medias y su árbol de jerarquía en la aplicación y la segunda tiene como responsabilidad la gestión de la información persistente, que antes sobrecargaba a la clase *Modelo* del patrón MVC original. Esta variación a su vez sustenta las características actuales de los sistemas multimedia como son: comunicación con bases de datos, archivos XML, o sistemas externos. (Consultar Anexo1).

1.6 Experiencias anteriores vinculadas al problema

Ámbito internacional

Desde los años noventa, el Centro Nacional de Información y Comunicación Educativa (CNICE) del Ministerio de Educación y Ciencia, ha llevado a cabo acciones encaminadas a

la producción de contenidos educativos digitales. Surgidos para dar respuesta a la creciente demanda de las aulas dotadas de servicios e infraestructuras tecnológicas, responden esencialmente al objetivo de apoyar los procesos de enseñanza y aprendizaje en la educación no universitaria.

Desde un primer momento se han caracterizado por tratarse de contenidos educativos abiertos, adaptables, modulares y flexibles. Desarrollados de forma agregada (con una estructura básica de bloques y módulos temáticos) en torno a un área y nivel educativo, y con tres perfiles de posibles usuarios (alumnado, profesorado y público), esta línea de trabajo ha marcado las pautas de la acción colaborativa llevada a cabo por todas las administraciones educativas en el marco del plan «Internet en la Escuela» (2002-2005).

«Internet en la Escuela»

«Internet en la Escuela» supuso un hito en el desarrollo de contenidos educativos digitales, tanto por el sistema de gestión de la producción, como por el establecimiento de un escenario de colaboración entre administraciones, pues garantizaba la existencia de un espacio para la reflexión, la innovación y la experimentación.

Su ámbito de definición se centró fundamentalmente en educación infantil y educación primaria, completando también la oferta de contenidos educativos para educación secundaria obligatoria y bachillerato, que hasta el momento habían constituido el objetivo esencial de los materiales producidos por el CNICE.

«Internet en la Escuela» ha definido las líneas fundamentales de la actual producción de contenidos educativos digitales, orientando sobre las formas, tiempos y usos que consolidarán éstos, no sólo como una alternativa, sino como una realidad diaria en los procesos de enseñanza aprendizaje en la educación no universitaria.

El plan «Internet en el Aula» (2005-2008)

Entroncado con los pasos anteriores, las administraciones educativas españolas, en el marco del desarrollo de la sociedad del conocimiento, han emprendido un nuevo e interesante camino en relación con la puesta a disposición de contenidos educativos digitales.

El plan «Internet en el Aula» (2005-2008) aborda la inclusión digital en el ámbito educativo con procesos de producción basados en sistemas de trabajo colaborativo, en los que los contenidos digitales abiertos son, a un tiempo, medio y fin.

Desde una mirada a los contenidos educativos digitales como fin, como producto final, el plan «Internet en el Aula» significa, de inicio, un crecimiento de los objetivos definidos hasta hoy en el fomento de su elaboración, difusión y utilización. Las administraciones e instituciones que desarrollan este plan –los Ministerios de Educación y Ciencia, de Industria, Turismo y Comercio, la entidad pública empresarial Red.es y las comunidades autónomas, se nutren de la experiencia del anterior programa «Internet en la Escuela» para abundar en la idea de unos contenidos digitales educativos de calidad, útiles, accesibles, modulares, interoperables y reutilizables, que no sólo faciliten la optimización de la producción conjunta de materiales, sino que además completen la oferta pública existente; permitan compartir el conocimiento y las soluciones; fomenten e impulsen la innovación educativa, e impliquen a cada vez más miembros de la comunidad educativa, tanto en su utilización como en la adaptación y en la generación de nuevos contenidos a partir de los existentes.

Ambos proyectos se basaron en el desarrollo de una arquitectura modular de jerarquía creciente (basada en modelos de agregación), donde los niveles de agregación definen y organizan la granularidad estructural y funcional de un ODE sobre la base de tres variables: la estructura, la funcionalidad, y la cobertura curricular. Finalizadas las fases previas, se ha seleccionado una iniciativa de estandarización que se adapta a las necesidades del proyecto: se trata del estándar learning object metadata (LOM), del Learning Technology Standards Committee (LTSC). A partir de aquí, se ha aceptado por consenso el diseño y elaboración de un perfil de aplicación específico de metadatos (LOMES), que contemple y satisfaga las necesidades específicas de los programas institucionales y de la comunidad educativa de destino. Concretamente, el perfil de aplicación español de forma consensuada entre las administraciones anteriormente citadas ha puesto especial atención en la organización de la metainformación educativa y didáctica.

1.7 Actualidad en la modelación del SWE en Cuba

Dada las características de surgimiento del software educativo, y que los primeros especialistas interesados en utilizar la computadora para la educación fueron precisamente los educadores; se utilizaron en dicho surgimiento un conjunto de recursos extrapolados de otras industrias o actividades, como lo fue el *Guión*, el cual luego se dividió en *de contenido* y *técnico*, tal y como sucede en la televisión; siendo todavía hoy el principal artefacto de representación de este tipo de aplicaciones.

Se pueden emplear diferentes formatos o técnicas para elaborarlo, pero en esencia todos deben tener una estructuración por *escenas* y cada una con un determinado nivel de detalle. Los guiones además no son más que una *breve sinopsis o descripción general* y

esquemática de cómo estará constituida la obra, acompañada de la estructura orgánica y la funcionalidad de cada elemento. La mayoría de los guiones contemplan como partes o componentes un *Esquema de escenas* y un *Grafo de navegación*. Otros equipos desarrolladores apoyan el trabajo de creación de los software educativos con artefactos como: *listado de medias* (textos, imágenes, videos, animaciones y sonidos) a utilizar en la aplicación educativa y *Documentación de los parlamentos* si se utiliza una mascota de guía en el software.

Es de notar con claridad como en la descripción ofrecida sobre los guiones multimedia, se distingue la recarga considerable del mismo en los aspectos que desde el punto de vista visual o de interés para el usuario en una aplicación educativa existen. Sin embargo es de difícil consecuencia la generación de una documentación de interés ingenieril a partir de un guión de este tipo.

En la última década se ha avanzado en este campo, logrando en la modelación de entornos educativos, la incorporación de lenguajes notacionales de propósito general como UML, el cual incorpora los elementos establecidos OCL, o sus extensiones como OMMMA – L. No obstante estos lenguajes no logran denotar todos los elementos, tanto ingenieriles como pedagógicos de las aplicaciones educativas cubanas en sus gráficos y semánticas utilizadas.

1.8 Situación Problemática

La Universidad de las Ciencias Informáticas (UCI) es una universidad que surgió como un nuevo proyecto de la Revolución Cubana, desempeñando un papel importante en el desarrollo de la Industria Cubana del Software (ICSW), y en la materialización de los proyectos asociados al programa cubano de informatización.

Hoy día se ha generado una fuerte línea de desarrollo de Productos de Software Educativo tales como:

- Productos de software educativo para Venezuela.
- Desarrollo de Colecciones para las escuelas cubanas.

Gran parte de estos productos son desarrollados por el Ministerio de Educación, como ejemplo podemos citar la colección primaria '*Mutisaber*'. En la actualidad en sus procesos de desarrollo se hace un breve análisis de su arquitectura o simplemente no se hace ninguno se lleva todo un proceso centrado en "***códigos de programación***", lo que trae como consecuencia que los productos no se entregan en el tiempo requerido y en algunos casos

inconformidades de los clientes. Por eso surge la necesidad de un estudio de los elementos estructurales en el proceso y concepción pedagógico del software educativo en esta institución. Para esto se hizo una investigación exhaustiva sobre los elementos estructurales en el proceso y concepción pedagógica del software educativo a nivel internacional, se tomaron aquellos aspectos que respondían al cumplimiento de las características del proceso de enseñanza-aprendizaje cubano, y se adaptaron al contexto productivo del MINED.

1.9 Conclusiones

A lo largo de este capítulo se han ofrecido los elementos teóricos que sirven de sustento científico a la investigación, así como se ha ofrecido cada uno de los aspectos a tener en cuenta en la situación problemática que genera el problema científico razón de esta investigación. El análisis de cada uno de estos aspectos mencionados permite arribar de manera parcial a las siguientes conclusiones:

- Para la producción de software educativo, en los últimos tiempos se ha incrementado mucho la utilización de la tecnología multimedia, ya que brinda muchas ventajas para lograr los objetivos perseguidos en un software educativo.
- Las buenas prácticas de la arquitectura de software aplicadas durante el ciclo de vida, tienen el potencial de incrementar la facilidad para entender un sistema de software y de desarrollar procesos para crearlo, asegurando que cualidades particulares y relevantes sean alcanzadas, y que se reduzca el costo.
- Se han desarrollado a lo largo de los años, desde la presentación del MVC a la comunidad científica variantes fundamentales, como el *MVC_E*, que en la actualidad logra un mayor acercamiento al SWE cubano.
- En la actualidad en el MINED se realiza un proceso de desarrollo del software que no se basa en un análisis previo de arquitectura.

2

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

2.1 Introducción

Antes de llegar a la solución propuesta en el presente capítulo se hace necesario destacar algunos aspectos del software educativo producido para el MINED como son: las características y sus elementos estructurales.

2.2 Proceso de desarrollo del software educativo

El diseño y producción de un software educativo multimedia, es tarea de un grupo multidisciplinario de profesionales. Si se pretende alcanzar los estándares de calidad que exige el mercado, es necesario prestar mucha atención a la hora de seleccionar los equipos de trabajos que se integrarán a tal efecto.

En general los especialistas que intervienen en este trabajo son: pedagogos, psicólogos, expertos en la materia a tratar, informáticos, guionistas, diseñadores, didactas, técnicos en audio y vídeo, entre otros.

Entre los grupos de trabajo que se suelen crear los asesores del MINED están los destinados a:

- **Elaboración del guión:** Prepara el guión detallado para el trabajo del resto de los equipos. Se integra con profesores de la asignatura cuyos contenidos serán expuestos, psicólogos, pedagogos, diseñadores y especialistas en informática educativa fundamentalmente.
- **Gestión de recursos multimedia:** Tiene como misión la búsqueda de todos los elementos a incluir en el software, esto es: textos, fotos, sonidos, vídeos, etc. y se responsabiliza con la obtención de las licencias de uso de los mismos. En general se integran a él, especialistas en audiovisuales y profesores.

- **Procesamiento de la información:** Su responsabilidad es dar el acabado final a la información, atendiendo a los formatos y especificaciones acordadas en el guión. En este equipo por lo general trabajan informáticos y especialistas en el procesamiento de los diferentes componentes multimedia.
- **Programación:** Realiza el ensamblaje definitivo de todos los elementos que conforman el software, garantizando que éste tenga la apariencia y funcionalidad previstas en el guión. Informáticos e infografistas tienen a su cargo estas tareas.
- **Realización de las pruebas:** Su función radica en controlar la calidad del software durante todo el proceso de desarrollo. Aplican las pruebas que se diseñan para medir la confiabilidad del software tanto desde el punto de vista conceptual como de su utilización. En este equipo figuran informáticos y profesores.

Todos los grupos anteriormente mencionados deben estar bajo la dirección del jefe del proyecto, quien velará por el buen funcionamiento de los mismos y porque existan vínculos de trabajos entre ellos que garanticen un trabajo coherente y el cumplimiento del cronograma.

2.3 Notaciones en la producción del SWE cubano

El guión

“La función primordial del guión es dar objetividad al proyecto de la obra multimedia de forma tal que pueda independizarse el proceso de ejecución del proyecto de concepción y diseño. (...) En segundo lugar, el guión es imprescindible para lograr una comunicación clara y precisa entre los integrantes del equipo de trabajo de manera que aunque realicen tareas independientes todos conozcan cómo tienen que hacer su labor y cómo encaja cada uno dentro de la obra.” (Barrera Yanes, 1998)

Otros autores hacen una clasificación más sencilla siguiendo el esquema clásico de las producciones audiovisuales. A diferencia de éstas, en el guión multimedia:

“(...) se recogerán de manera exhaustiva los elementos que han de intervenir en cada una de las pantallas o secuencias, las acciones que se desarrollarán, el grafismo utilizado y el “tempo” que mantendrán, es decir, el orden de representación y el tiempo del mismo. También se debe describir detalladamente cuándo sucederá, bien por la intervención directa del usuario bien por otro tipo de causas, como por ejemplo la ausencia de acciones por parte de éste”. (Pérez Huertas, 1998)

Árboles o mapas de navegación.

Un mapa de navegación es la representación gráfica de la organización de la información. Expresa todas las relaciones de jerarquía y secuencia y permite elaborar escenarios de comportamiento de los usuarios. También grafica, de modo que todos los profesionales participantes en un proyecto lo tengan claro, diferencias entre páginas dinámicas, administrables o estáticas.

El principal valor de un mapa de navegación es que permite anticipar errores de organización de la información, de modo de corregirlos cuando aún no se ha invertido tiempo y dinero en la construcción del producto.

2.4 Características del software educativo en el MINED

Es necesario primeramente hacer un recorrido por las características, estructura básica, funciones y la clasificación de los programas didácticos en general para tener en cuenta cuales son las características que presenta el software educativo producido en la UCI, para la institución del MINED.

2.5 Tipologías del Software Educativo

Existen diversas tipologías de software educativo (cada una con fines diferentes), las cuales trataremos más adelante, pero todos los software educativos comparten 5 características esenciales. (Marqués 1996)

- Son materiales elaborados con una finalidad didáctica.
- Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- Son Interactivos, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- Individualizan el trabajo de los estudiantes, puesto que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- Son fáciles de usar, debido a que los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos electrónicos necesarios para usar un video, es decir son mínimos, aunque cada programa tiene reglas de funcionamiento que es necesario conocer.

2.6 Estructura básica de los programas didácticos

Muchos de los programas didácticos, al igual que la mayoría de los programas informáticos sin ninguna finalidad educativa, tienen tres módulos principales definidos claramente: el que gestiona la comunicación con el usuario (Sistema input/output), el que contiene debidamente organizados los contenidos informáticos del programa (BD) y el que tramita las actuaciones del ordenador y sus respuestas a las acciones de los usuarios (motor). Ver explicación detallada en el Anexo 2.

2.7 Funcionalidades del Software Educativo

Los programas didácticos cuando se aplican a la realidad educativa, realizan las funciones básicas propias de los medios didácticos en general. (Marqués 1996)

No se puede afirmar que el software educativo por si mismo sea bueno o malo, todo dependerá del uso que se le de, y de la manera en que se utilice en cada situación concreta. Por último la funcionalidad, las ventajas e inconvenientes que pueda tener el uso de los materiales didácticos, será el resultado de las características del material, de su adecuación al contexto educativo al que se aplica y de la manera en que el profesor organice su utilización (si el profesor no organiza bien su utilización de manera que se logre los objetivos que con el software se pretende, los cuales deben ser similares a los objetivos pedagógicos del contexto al que se aplica, el software no cumple con los requisitos funcionales explícitamente establecidos).

Entre los elementos funcionales del software educativo podemos citar:

Originalidad y uso de la tecnología avanzada: Los programas didácticos deben presentar entornos originales, bien diferenciados de otros materiales didácticos y utilizar las crecientes potencialidades del ordenador y de las tecnologías multimedia e hipertexto en general, de forma tal que el ordenador resulte intrínsecamente potenciador del proceso de aprendizaje, favorezca la asociación de ideas y creatividad, permita la practica de nuevas técnicas, la reducción del tiempo y del esfuerzo necesario para aprender y facilite aprendizajes más completos y significativos. La inversión financiera, intelectual y metodológica que supone elaborar un programa educativo, solo se justifica si el ordenador mejora lo que ya existe.

Facilidad de uso e instalación: Los programas educativos deben ganarse la atención de los estudiantes, para esto es necesario que sean agradables, fáciles de usar y autoexplicativos, de forma tal que los estudiantes puedan utilizarlos sin tener que pasar por un curso o tener que realizar una exhaustiva lectura de los manuales para aprender a

trabajar con el software. Agreguemos a esto, que el software no debe tener largas tareas previas de configuración y además, debe posibilitarle al estudiante conocer en todo momento el lugar del programa donde se encuentra y las opciones a su alcance para moverse según sus preferencias (retroceder, avanzar o ir al índice).

Adaptación a diversos contextos: Debido a que los materiales didácticos deben dar una buena respuesta a las diversas necesidades educativas de sus destinatarios y puedan utilizarse de múltiples maneras según las circunstancias, es conveniente que tengan una alta capacidad de adaptación a:

- **Entornos de uso:** Pueden ser aulas de informática, clases con un único ordenador, clases con pizarra electrónica (en un futuro).
- **Agrupamientos:** Por trabajo individual, grupo cooperativo o competitivo.
- **Estrategias didácticas:** Ya sea enseñanza dirigida, exploración guiada, y libre descubrimiento.
- **Usuarios y contextos formativos:** Estilos de aprendizajes, circunstancias culturales y necesidades formativas, problemáticas para el acceso a la información (visual y matriz).

Para lograr esta adaptación, los materiales didácticos en soporte informático deberán tener ciertas características que lo permita, como las que siguen a continuación:

- **Programables**, para poder modificar algunos parámetros como: nivel de dificultad, tiempo de respuesta, números de usuarios simultáneos e idioma.
- **Abiertos**, permitiéndole al profesorado modificar fácilmente los contenidos de las bases de datos y las actividades que proporcionarán a los estudiantes.
- **Facilitar la impresión de los contenidos**, sin una excesiva fragmentación. .. Incluir un sistema de evaluación y seguimiento (control), que proporcione informes de las actividades realizadas por los estudiantes: temas tratados, nivel de dificultad, tiempo invertido, errores que ha cometido, itinerarios seguidos para resolver los problemas y otros.
- **Permitir** el seguimiento de los trabajos empezados con anterioridad.
- **Promover** el desarrollo de actividades complementarias (individuales o en grupo) con el uso de otros materiales (fichas, diccionarios o libros).
- **Dar respuesta** a las problemáticas de acceso de los colectivos, con necesidades especiales, proporcionando interfaces ajustables según las características de los usuarios (tamaño de letra, uso del teclado, ratón o periféricos adaptativos).

2.8 Tipologías de los Programas Didácticos

Los programas didácticos, aunque presentan rasgos esenciales básicos y una estructura general común, se presentan de diversas formas. Con el fin de establecer un orden a esta disparidad se han elaborados múltiples tipologías que clasifican los programas didácticos a partir de diferentes criterios. Unos de estos criterios se basa en la consideración del tratamiento de los errores que cometen los estudiantes, distinguiéndose (Marqués 1996):

- **Programas de tutoriales directivos**, que hacen preguntas a los estudiantes y controlan en todo momento su actividad. El ordenador adopta el papel de juez poseedor de la verdad y examina al alumno. Se producen errores cuando la respuesta del alumno está en desacuerdo con la que el ordenador tiene como correcta. El error lleva implícita la noción del fracaso.
- **Programas no directivos**, en los que el ordenador adopta el papel de un laboratorio o instrumento a disposición de la iniciativa de un alumno que pregunta y tiene una libertad de acción sólo limitada por las normas del programa. El ordenador no juzga las acciones del alumno, se limita a procesar los datos que éste introduce y a mostrar las consecuencias de sus acciones sobre un entorno. Objetivamente no se producen errores, sólo desacuerdos entre los efectos esperados por el alumno y los efectos reales de sus acciones sobre el entorno. No está implícita la noción de fracaso. El error es sencillamente una hipótesis de trabajo que no se ha verificado y que se debe sustituir por otra. En general, siguen un modelo pedagógico de inspiración cognitivista, potencian el aprendizaje a través de la exploración, favorecen la reflexión y el pensamiento crítico y propician la utilización del método científico.

Otra clasificación interesante de los programas tiene en cuenta la posibilidad de modificar los contenidos del programa y distingue entre programas cerrados (que no pueden modificarse) y abiertos (que pueden modificarse), que proporcionan un esqueleto, una estructura, sobre la cual los alumnos y los profesores pueden añadir el contenido que les interese. De esta manera se facilita su adecuación a los diversos contextos educativos y permite un mejor tratamiento de la diversidad de los estudiantes.

Pero la clasificación que tiene en cuenta el grado de control del programa sobre las actividades de los alumnos y la estructura de su algoritmo, es la que proporciona categorías más claras y útiles a los profesores, por ende, es la que se procede a dar una breve explicación a continuación.

Programas Tutoriales:

Son programas que en mayor o menor medida dirigen y tutorizan, el trabajo de los alumnos. Pretenden que, a partir de unas informaciones y mediante la realización de ciertas actividades previstas de antemano, los estudiantes pongan en juego determinadas capacidades y aprendan o refuercen unos conocimientos y/o habilidades. Cuando se limitan a proponer ejercicios de refuerzo sin proporcionar explicaciones conceptuales previas, se denominan programas tutoriales de ejercitación, como es el caso de los programas de preguntas (drill&practice, test) y de los programas de adiestramiento psicomotor, que desarrollan la coordinación neuromotriz en actividades relacionadas con el dibujo, la escritura y otras habilidades psicomotrices.

En cualquier caso, son programas basados en los planteamientos conductistas de la enseñanza, que comparan las respuestas de los alumnos con los patrones que tienen como correctos, guían el aprendizaje de los estudiantes y facilitan la realización de prácticas más o menos rutinarias y su evaluación; en algunos casos una evaluación negativa genera una nueva serie de ejercicios de repaso. A partir de la estructura del algoritmo de los programas, se distinguen cuatro categorías:

- **Programas lineales**, que presentan al alumno una secuencia de información y/o ejercicios (siempre la misma o determinada aleatoriamente) con independencia de la corrección o incorrección de sus respuestas. Herederos de la enseñanza programada, transforman el ordenador en una máquina de enseñar transmisora de conocimientos y adiestradora de habilidades. No obstante, su interactividad resulta pobre y el programa se hace largo de recorrer.
- **Programas ramificados**, basados inicialmente también en modelos conductistas, siguen recorridos pedagógicos diferentes según el juicio que hace el ordenador sobre la corrección de las respuestas de los alumnos o según su decisión de profundizar más en ciertos temas. Ofrecen mayor interacción, más opciones, pero la organización de la materia suele estar menos compartimentada que en los programas lineales y exigen un esfuerzo más grande al alumno. Pertenecen a éste grupo los programas multinivel, que estructuran los contenidos en niveles de dificultad y previenen diversos caminos, y los programas ramificados con dientes de sierra, que establecen una diferenciación entre los conceptos y las preguntas de profundización, que son opcionales.
- **Entornos tutoriales**, en general, están inspirados en modelos pedagógicos cognitivistas, y proporcionan a los alumnos una serie de herramientas de búsqueda y de proceso de la información que pueden utilizar libremente para construir la respuesta a las preguntas del programa. Este es el caso de los entornos de resolución de problemas,

"problem solving", donde los estudiantes conocen parcialmente las informaciones necesarias para su resolución y han de buscar la información que falta y aplicar reglas, leyes y operaciones para encontrar la solución. En algunos casos, el programa no sólo comprueba la corrección del resultado, sino que también, tiene en cuenta la idoneidad del camino que se ha seguido en la resolución. Sin llegar a estos niveles de análisis de las respuestas, podemos citar como ejemplo de entorno de resolución de problemas el programa MICROLAB DE ELECTRÓNICA.

- **Sistemas tutoriales expertos**, como los Sistemas Tutores Inteligentes (Intelligent Tutoring Systems), que, elaborados con las técnicas de la Inteligencia Artificial y teniendo en cuenta las teorías cognitivas sobre el aprendizaje, tienden a reproducir un diálogo auténtico entre el programa y el estudiante, y pretenden comportarse como lo haría un tutor humano: guían a los alumnos paso a paso en su proceso de aprendizaje, analizan su estilo de aprender y sus errores y proporcionan en cada caso la explicación o ejercicio más conveniente.

Base de Datos:

Proporcionan datos organizados, en un entorno estático, según determinados criterios, y facilitan su exploración y **consulta** selectiva. Se pueden emplear en múltiples actividades como por ejemplo: seleccionar datos relevantes para resolver problemas, analizar y relacionar datos, extraer conclusiones, y comprobar hipótesis. Las preguntas que acostumbran a realizar los alumnos son del tipo: **¿Qué características tiene este dato?** **¿Qué datos hay con la característica X?** **¿Qué datos hay con las características X y Y?** Las bases de datos pueden tener una estructura jerárquica (si existen unos elementos subordinantes de los que dependen otros subordinados, como los organigramas), relacional (si están organizadas mediante unas fichas o registros con una misma estructura y rango) o documental (si utiliza descriptores y su finalidad es almacenar grandes volúmenes de información documental: revistas, periódicos). En cualquier caso, según la forma de acceder a la información se pueden distinguir dos tipos:

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

Simuladores:

Presentan un modelo o entorno dinámico (generalmente a través de gráficos o animaciones interactivas), facilitan su exploración y modificación a los alumnos, que pueden realizar aprendizajes inductivos o deductivos, mediante la observación y la manipulación de la estructura subyacente; de esta manera pueden descubrir los elementos del modelo, sus interrelaciones, pueden tomar decisiones y adquirir experiencia directa delante de unas situaciones que frecuentemente resultarían difícilmente accesibles a la realidad (control de una central nuclear, contracción del tiempo, pilotaje de un avión). También se pueden considerar simulaciones ciertos videojuegos que, al margen de otras consideraciones sobre los valores que incorporan (generalmente no muy positivos) facilitan el desarrollo de los reflejos, la percepción visual y la coordinación psicomotriz en general, además de estimular la capacidad de interpretación y de reacción ante un medio concreto.

En cualquier caso, **posibilitan un aprendizaje significativo por descubrimiento** y la investigación de los estudiantes/experimentadores puede realizarse en tiempo real o en tiempo acelerado, según el simulador, mediante preguntas del tipo: **¿Qué pasa al modelo si modifico el valor de la variable X? ¿Y si modifico el parámetro Y?** Se pueden diferenciar dos tipos de simuladores:

- **Modelos físico-matemáticos**, presentan de manera numérica o gráfica una realidad que tiene unas leyes representadas por un sistema de ecuaciones deterministas. Se incluyen aquí los programas-laboratorio, algunos trazadores de funciones y los programas que mediante un convertidor analógico-digital captan datos analógicos de un fenómeno externo al ordenador y presentan en pantalla un modelo del fenómeno estudiado o informaciones y gráficos que van asociados. Estos programas a veces son utilizados por profesores delante de la clase a manera de pizarra electrónica, como demostración o para ilustrar un concepto, facilitando así la transmisión de información a los alumnos, que después podrán repasar el tema interactuando con el programa.
- **Entornos sociales**, presentan una realidad regida por unas leyes no del todo deterministas. Se incluyen aquí los juegos de estrategia y de aventura, que exigen una destreza cambiante a lo largo del tiempo.

Constructores:

Son programas que tienen un entorno programable. Facilitan a los usuarios unos elementos simples con los cuales pueden construir elementos o entornos más complejos. De esta manera potencian el aprendizaje heurístico y, de acuerdo con las **teorías cognitivistas**, facilitan a los alumnos la construcción de sus propios aprendizajes, que surgirán a través de la reflexión que realizarán al diseñar programas y comprobar inmediatamente cuando los

ejecuten, la relevancia de sus ideas. El proceso de creación que realiza el alumno genera preguntas del tipo: *¿Qué sucede si añado o elimino el elemento X?* Se pueden distinguir dos tipos de constructores:

- **Constructores específicos**, ponen a disposición de los estudiantes una serie de mecanismos de actuación (generalmente en forma de órdenes específicas) que les permiten llevar a cabo operaciones de un cierto grado de complejidad, mediante la construcción de determinados entornos, modelos o estructuras, y de esta manera avanzan en el conocimiento de una disciplina o entorno específico.
- **Lenguajes de programación**, como LOGO, PASCAL y BASIC, que ofrecen unos "laboratorios simbólicos" en los que se pueden construir un número ilimitado de entornos. Aquí los alumnos se convierten en profesores del ordenador. Además, con las interfaces convenientes, pueden controlar pequeños robots contruidos con componentes convencionales (arquitecturas, motores), de manera que sus posibilidades educativas se ven ampliadas incluso en campos pre-tecnológicos. Así los alumnos pasan de un manejo abstracto de los conocimientos con el ordenador a una manipulación concreta y práctica en un entorno informatizado, que facilita la representación y comprensión del espacio y la previsión de los movimientos.

Dentro de este grupo de programas hay que destacar el lenguaje LOGO, creado en 1969 para Seymour Papert, que constituye el programa didáctico más utilizado en todo el mundo. LOGO es un programa constructor que tiene una doble dimensión:

- Proporciona **entornos de exploración** donde el alumno puede experimentar y comprobar las consecuencias de sus acciones, de manera que va construyendo un marco de referencia y esquemas de conocimiento, que facilitarán la posterior adquisición de nuevos conocimientos.
- Facilita una actividad formal y compleja, próxima al terreno de la construcción de estrategias de resolución de problemas: **la programación**. A través de ella los alumnos pueden establecer proyectos, tomar decisiones y evaluar los resultados de sus acciones.

Programas Herramientas:

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

de la definición que se ha dado de software educativo. No obstante, se han elaborado algunas versiones de estos programas "para niños" que limitan sus posibilidades a cambio de una, no siempre clara, mayor facilidad de uso. De hecho, muchas de estas versiones resultan innecesarias, ya que el uso de estos programas cada vez resulta más sencillo y cuando los estudiantes necesitan utilizarlos o su uso les resulta funcional, aprenden a manejarlos sin dificultad. Los programas más utilizados de este grupo son:

- **Procesadores de textos**, son programas que, con la ayuda de una impresora, convierten el ordenador en una fabulosa máquina de escribir. En el ámbito educativo debe hacerse una introducción gradual que puede empezar a lo largo de la Enseñanza Primaria, y ha de permitir a los alumnos familiarizarse con el teclado y con el ordenador en general, y sustituir parcialmente la libreta de redacciones por un disco (donde almacenarán sus trabajos). Al escribir con los procesadores de textos los estudiantes pueden concentrarse en el contenido de las redacciones y el resto de los trabajos que tengan encomendados despreocupándose por la caligrafía. Además el corrector ortográfico, que suelen incorporar, les ayudará a revisar las posibles faltas de ortografía antes de entregar el trabajo.

Además de este empleo instrumental, los procesadores de textos permiten realizar múltiples actividades didácticas, por ejemplo:

- Ordenar párrafos, versos, estrofas.
- Insertar frases y completar textos.
- Separar dos poemas.
- **Gestores de bases de datos**, sirven para generar potentes sistemas de archivo ya que permiten almacenar información de manera organizada y posteriormente recuperarla y modificarla. Entre las muchas actividades con valor educativo que se pueden realizar están las siguientes:
 - Revisar una base de datos ya construida para buscar determinadas informaciones y recuperarlas.
 - Recoger información, estructurarla y construir una nueva base de datos.
- **Hojas de cálculo**, son programas que convierten el ordenador en una versátil y rápida calculadora programable, facilitando la realización de actividades que requieran efectuar muchos cálculos matemáticos. Entre las actividades didácticas que se pueden realizar con las hojas de cálculo están las siguientes:

- Aplicar hojas de cálculo ya programadas a la resolución de problemas de diversas asignaturas, evitando así la realización de pesados cálculos y ahorrando un tiempo que se puede dedicar a analizar los resultados de los problemas.
- Programar una nueva hoja de cálculo, lo que exigirá previamente adquirir un conocimiento preciso del modelo matemático que tiene que utilizar.
- **Editores gráficos**, se emplean desde un punto de vista instrumental para realizar dibujos, portadas para los trabajos, murales y anuncios. Además, constituyen un recurso idóneo para desarrollar parte del currículum de Educación Artística: dibujo, composición artística y uso del color entre otros.
- **Programas de comunicaciones**, son programas que permiten que ordenadores lejanos (si disponen de módem) se comuniquen entre sí, a través de las líneas telefónicas y puedan enviarse mensajes, gráficos y programas. Desde una perspectiva educativa estos sistemas abren un gran abanico de actividades posibles para los alumnos, por ejemplo:
 - Comunicarse con otros compañeros e intercambiarse informaciones.
 - Acceder a bases de datos lejanas para buscar determinadas informaciones.
- **Programas de experimentación asistida**, a través de variados instrumentos y convertidores analógico-digitales, recogen datos sobre el comportamiento de las variables que inciden en determinados fenómenos. Posteriormente con estas informaciones se podrán construir tablas y elaborar representaciones gráficas que representen relaciones significativas entre las variables estudiadas.
- **Lenguajes y sistemas de autor**, son programas que facilitan la elaboración de programas tutoriales a los profesores que no disponen de grandes conocimientos informáticos. Utilizan unas pocas instrucciones básicas que se pueden aprender en pocas sesiones. Algunos incluso, permiten controlar vídeos y dan facilidades para crear gráficos y efectos musicales, de manera que pueden generar aplicaciones multimedia. Algunos de los más utilizados en entornos PC han sido: PILOT, PRIVATE TUTOR, TOP CLASS, LINK WAY y QUESTION MARK.

Después de exponer esta serie de aspectos significativos sobre el software educativo, se puede enmarcar el entorno del Software Educativo a nivel internacional como se representa en la Figura 7 que se muestra a continuación:

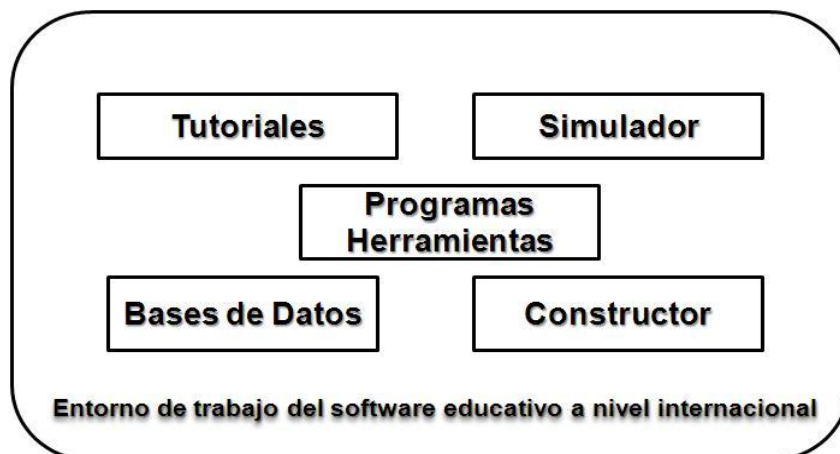


Figura 7 Entorno de trabajo del software educativo a nivel internacional

2.9. Características del SWE en el MINED

Teniendo en cuenta las características del proceso de enseñanza-aprendizaje cubano (Anexo 3) y los resultados obtenidos al aplicar las entrevistas (Anexo 4) y el método histórico-lógico, se llega a la conclusión, que el software educativo producido para el MINED es mixto (ver Figura 8) conteniendo elementos de:

- **Tutoriales:** Son programas que en mayor o menor medida dirigen, tutorizan el trabajo de los alumnos. En cualquier caso, son programas basados en los planteamientos conductistas de la enseñanza, que comparan las respuestas de los alumnos con los patrones que tienen como correctos, guían los aprendizajes de los estudiantes y facilitan la realización de prácticas más o menos rutinarias y su evaluación; en algunos casos una evaluación negativa genera una nueva serie de ejercicios de repaso. A partir de la estructura de su algoritmo, se distinguen cuatro categorías (Programas Lineales, Ramificado, Entornos Tutoriales y Sistema Tutoriales Expertos).
- **Base de Datos:** Proporcionan datos organizados en un entorno estático, según determinados criterios y facilitan su exploración y consulta selectiva. Pueden tener una estructura jerárquica, relacional o documental. En cualquier caso, según la forma de acceder a la información se pueden distinguir dos tipos de simulador (Base de Datos Convencionales y Base de Datos Tipo Experto).
- **Constructor:** Son programas con un entorno programable, que facilitan a los usuarios elementos simples, con los cuales pueden construir elementos o entornos más complejos. Así, potencian el aprendizaje heurístico y facilitan a los alumnos la construcción de sus propios aprendizajes, de acuerdo con las teorías cognitivistas.

Se pueden distinguir dos tipos de constructores (Constructores Específicos y Lenguajes de Programación).

- **Simulador:** Presentan un modelo o entorno dinámico (generalmente a través de gráficos o animaciones interactivas), facilitan su exploración y modificación a los alumnos que pueden realizar aprendizajes inductivos o deductivos, mediante la observación y la manipulación de la estructura subyacente. Estos programas didácticos posibilitan un aprendizaje significativo por descubrimiento y la investigación de los estudiantes/experimentadores pueden realizarse en tiempo real o en tiempo acelerado, según el simulador. Se pueden distinguir dos tipos de simulador (Modelo Físico-Matemático y Entorno Sociales).

Lo que hace que sea un software complejo.

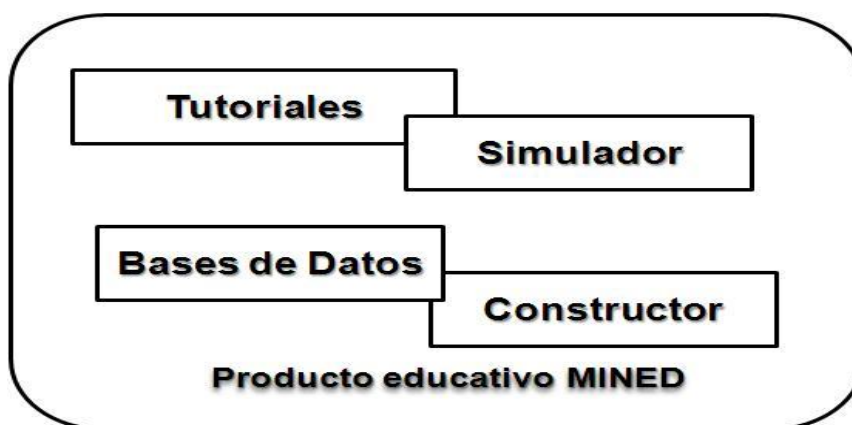


Figura 8 Entorno del software educativo producido para el MINED

2.10 Elementos Estructurales del software educativo MINED.

La informática encontró una buena vía de acceso a los hogares y fue por medio de *la multimedia*. Esta evolución en los computadores domésticos, ha hecho lo que hasta hace un par de años era una aburrida máquina de proceso de texto y archivo de datos se haya convertido en una excelente máquina con capacidad de mostrar video y sonido al mismo tiempo y con calidad de *compact disc* en un mismo aparato. La multimedia también sirve como un medio educativo, cultural para los niños; actualmente existen colegios tanto primarios como secundarios que utilizan computadores como un medio de enseñanza y aprendizaje; ya sea tanto teórica como práctica; y para estos utilizan software que abarcan diversos temas, que comprenden desde la matemática, geografía, ciencia, artística, gramática y hasta inclusive música con ellos.

Los profesores se han dado cuenta de las grandes posibilidades que los CD-ROMs brindan en materia educativa: son obras cada día más completas que motivan por su gran número de estímulos, **el aprendizaje**.

“Multimedia estimula los ojos, oídos, yemas de los dedos y, lo más importante, la cabeza.”

*“Multimedia se compone, de combinaciones entrelazadas de elementos de **texto, gráfico, sonido, animación y vídeo, Diseño, Escenas y sistemas de navegación.**”*

A la hora de crear un producto multimedia los diseñadores tienen en cuenta varios aspectos:

- Consistencia: se basa en seguir una uniformidad en las entradas y salidas del sistema.
- Retroalimentación: es la información o respuesta que da la computadora a alguna acción o mandato que se hizo.
- Minimizar las posibilidades de error: Proveer al usuario solamente los comandos que son posibles ejecutarse bajo ciertas circunstancias es una buena manera de prevenir errores.
- Proveer recuperación de errores: Todos cometemos errores y no poder corregirlos afecta considerablemente en la productividad.
- Proveerle una ayuda al usuario (al presionar F1 o mediante un comando).
- Minimizar memorización: En algunos diseños de interfaces muchas veces se fuerza al usuario a memorizar mucha información que no es necesaria. Es importante involucrar al usuario en reconocer en lugar de memorizar.
- Acomodar múltiple niveles de habilidades: Los sistemas interactivos deben de ser diseñados para diferentes tipos de usuarios, desde el usuario nuevo e inexperto hasta el usuario más experto que deberá de trabajar por horas el sistema.
- No se deben colocar demasiados objetos en la pantalla, y los que existen deben estar bien distribuidos. Cada elemento visual influye en el usuario no sólo por sí mismo, sino también por su combinación con el resto de elementos presentes en la pantalla.
- Debe hacerse un uso adecuado de la terminología. Debe tenerse cuidado con las diferencias culturales (gestos, terminología, dibujos, formatos de teléfonos o calendarios, etc.).

Como se ha mencionado anteriormente en la Universidad de las Ciencias Informáticas se producen grandes cantidades de productos multimedia para la institución del MINED, dichos

productos, se construyen por una concepción pedagógica que se ha dado a conocer como Hiperentornos de Aprendizaje en la que se integran armónicamente por módulos, contemplando una interfaz estandarizada, que proporciona un ambiente de trabajo amigable e intuitivo con un alto nivel de interactividad para acceder a la información existente en el software.

2.11 Descripción de la solución propuesta

La arquitectura del software alude a “la estructura global del software y a las formas en que la estructura proporciona la integridad conceptual de un sistema”. En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa (módulos), la manera en que los componentes interactúan y la estructura de datos que van a utilizar los componentes. Sin embargo, en un sentido más amplio, los “componentes” se pueden generalizar para presentar los elementos principales del sistema y sus interacciones. (Pressman, 2002)

En este epígrafe se plantea, la solución propuesta que constituye una representación de los elementos arquitectónicos del software educativo que se produce en la Universidad de las Ciencias Informáticas para el MINED, la cual tiene como base al patrón **MVC** en su variante **MVC_E**, expuesto en el capítulo anterior, el mismo se centra en los aspectos pedagógicos, a la metodología y desarrollo de los software educativos cubanos. (Ver figura 9)

Como se ha planteado en epígrafes anteriores el software educativo cubano, tiene diferentes características marcadas en comparación con los desarrollados en otros países del mundo. Todo esto producido por el avance y la experiencia acumulada en el área de la pedagogía en nuestro país que lo ubica en uno de los primeros lugares en el planeta en este sentido. Las aplicaciones educativas cubanas explotan grandemente los conceptos del entorno hipermedia, así como incorporan de forma profunda las técnicas y conocimientos de bases de datos relacionales y la Programación Orientada a Objetos (P.O.O) más avanzada que se utiliza hoy día; como consecuencia de la necesidad de implementar diversos y complejos métodos pedagógicos desarrollados por nuestros educadores en las distintas ramas de la enseñanza cubana.

Si en otros países las multimedia llamadas educativas se circunscriben a meros software que presentan y evalúan alguna determinada materia, en Cuba por el contrario se desarrollan productos que realizan análisis exhaustivos de la navegación del usuario, mantienen actualizado el historial de trabajo de los usuarios y permiten preparar el software para diferentes entornos de trabajo a dichos usuarios teniendo en cuenta determinadas

características. Esto hace que nuestras aplicaciones se vean grandemente recargadas en el almacenamiento, procesamiento y actualización de una gran cantidad de información constantemente.

Al igual que cualquier programa didáctico, las aplicaciones multimedia producidas para el MINED tienen una estructura general común, sus componentes se agrupan en tres módulos principales : (1) el que gestiona la comunicación con el usuario, (2) el que contiene debidamente organizados los contenidos informáticos del programa (BD) y (3) el que tramita las actuaciones del ordenador y sus respuestas a las acciones de los usuarios (motor).

Al mismo tiempo es menester el análisis de las características del diseño de estas aplicaciones desde el punto de vista de sus clases y la arquitectura “¿Qué tipo de visibilidad deberían tener otros paquetes respecto a la capa de presentación? ¿Cómo deberían comunicarse con las ventanas las clases no relacionadas con ellas? Por lo regular conviene que no haya un acoplamiento directo de otros componentes con los objetos ventana porque éstos se encuentran relacionadas con una aplicación en particular, mientras que (en teoría) podemos reutilizar en nuevas aplicaciones los componentes sin acceso a ellas o unirlos a una nueva interfaz.

Teniendo en cuenta el creciente uso de la P.O.O. en la concepción e implementación de este tipo de software y la gran actualidad que tiene el uso de patrones internacionalmente aceptados para estas funciones, a si como su estructura general común, se propone al patrón **Modelo-Vista-Controlador-Entidad** para la concepción del diseño de las aplicaciones educativas para el MINED y en lo relacionado al trabajo con la información en las mismas. (Ver figura 10)

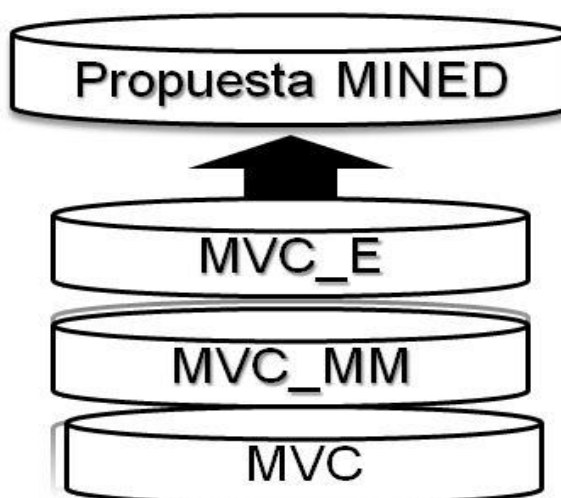


Figura 9 Solución propuesta para el MINED

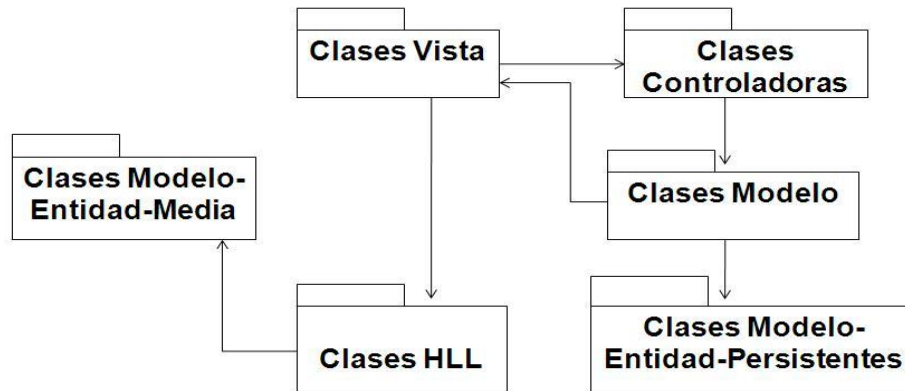


Figura 10 El MVC_E como patrón arquitectónicos del software educativo MINED

Para un mejor entendimiento de cómo quedarán estructurados los componentes en las futuras aplicaciones educativas para el MINED se explica a continuación:

- Clases **Vista** que recibirán las peticiones del usuario al sistema y mostrarán los mensajes de salida o respuestas.
- Clases **Controladoras** que gestionarán las peticiones y la muestra de las respuestas.
- Clases **Modelo** que contendrán la lógica de negocio para el procesamiento de la información.
- Clases **Modelo-Entidad-Persistente** para el procesamiento de la información persistente.
- Clases **Modelo-Entidad-Media** correspondientes a la representación del árbol jerárquico de medias.
- Clases correspondientes al **HLL** (*Lenguajes de Alto Nivel*) seleccionados para la programación del software.

Se presenta a continuación un resumen del Patrón **MVC_E** utilizando la plantilla estándar para la descripción de patrones:

Contexto/Problema:

Conviene desacoplar los objetos dominio (**Modelo**), las ventanas (**Vista**) y los manejadores (**Controlador**), a fin de brindar soporte a un mayor reuso de los objetos dominio y reducir al mínimo el impacto que los cambios de la interfaz y los manejadores tienen en ellos. ¿Qué hacer?

Solución:

Definir las clases dominio (**Modelo**) para que no tengan acoplamiento ni visibilidad directa respecto a las clases ventana (**Vista**) y para que los datos de la aplicación y de la funcionalidad se conserven en las clases de dominio, no en las de ventana. Definir las clases manejadores (**Controlador**) para que procesen los eventos (peticiones) al sistema y redirecciones a las clases dominio y ventana tanto el procesamiento como la visualización de resultados respectivamente.

Algunos de sus principales beneficios son:

- Menor acoplamiento.
 - Desacopla las vistas de los modelos.
 - Desacopla los modelos de la forma en que se muestran e ingresan los datos.
- Mayor cohesión
 - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).
- Las vistas proveen mayor flexibilidad y agilidad.
 - Se puede crear múltiples vistas de un modelo.
 - Las vistas pueden anidarse.
 - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
 - Se puede sincronizar las vistas.
 - Las vistas pueden concentrarse en diferentes aspectos del modelo.
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales
 - Una vista para cada dispositivo que puede variar según sus capacidades.
 - Una vista para la Web y otra para aplicaciones de escritorio.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

2.11 Conclusiones

En el capítulo que recién se ha presentado se hace un esbozo de los elementos estructurales que presenta el software educativo que se produce en el MINED, así como, las particulares que lo diferencia del SWE producido a nivel mundial. Al mismo tiempo se ha podido arribar a las siguientes conclusiones:

- El software educativo cubano, tiene diferentes características marcadas en comparación con los desarrollados en otros países del mundo.
- El SWE que se produce para el MINED tiene una estructura general común, sus componentes se agrupan en tres módulos principales:
 - El que gestiona la comunicación con el usuario.
 - El que contiene debidamente organizados los contenidos informáticos del programa (BD).
 - El que tramita las actuaciones del ordenador y sus respuestas a las acciones de los usuarios.
- Se propone al MVC_E como patrón arquitectónico para modelar el SWE que se produce para el MINED.

3

CAPÍTULO 3: EVALUACIÓN TÉCNICA DE LA PROPUESTA DE SOLUCIÓN

3.1 Introducción

En el presente capítulo se realiza la evaluación técnica de la propuesta descrita en el capítulo anterior. Para ello se utiliza el método Experto, el cual se basa en la evaluación cuantitativa de criterios previamente definidos por parte de expertos en el tema, que permite determinar si se acepta o no la propuesta analizada.

3.2 Guía para la evaluación técnica

A continuación se describen los pasos que se efectuaron para llevar a cabo la evaluación utilizando el método Experto:

1. Se elaboran los criterios que fueron utilizados en la evaluación y se agrupan de acuerdo a las características de la propuesta.

Grupo No 1: Criterios de mérito científico.

1. Valor científico de la propuesta.
2. Calidad de la investigación.
3. Contribución científica.

Grupo No 2: Criterios de implantación

4. Necesidad de empleo de la propuesta.
5. Obtención de productos finales con calidad.
6. Posibilidades de aplicación.

Grupo No 3: Criterios de flexibilidad.

7. Adaptabilidad a proyectos productivos de SWE independientemente del tipo de SWE que desarrollen.

8. Capacidad de la solución propuesta para adecuarse al proceso de desarrollo del SWE producido para el MINED.
9. Capacidad de la solución propuesta para adecuarse a los procesos de desarrollo de proyectos productivos que no desarrollen SWE.

Grupo No 4. Criterios de impacto.

10. Efectos en la mejora del proceso de desarrollo del SWE para el MINED.
11. Repercusión en la calidad de los procesos de desarrollo en los proyectos productivos de SWE para el MINED.

2. Se le asigna un peso relativo a cada grupo de criterios de acuerdo al porcentaje que representa cada grupo del total y los intereses a evaluar.

Grupo No. 1.....25

Grupo No. 2.....30

Grupo No.3.....25

Grupo No.4.....20

3. Se realiza una selección de 7 expertos en la cual se tiene en cuenta su especialidad, grado científico y currículum.
4. Se hace entrega de la propuesta que se desea validar a todos los expertos para que se documenten sobre el tema de la investigación y luego expresen sus criterios en sendos modelos. En el primero (Ver Anexo 5), los expertos conceden pesos a cada uno de los criterios establecidos, teniendo en cuenta que la suma de los valores dados para un grupo no exceda del peso relativo asignado a este.

El segundo modelo (Ver Anexo 6) permite realizar una evaluación cuantitativa de cada criterio con una escala de 1 a 5 y la apreciación cualitativa con una clasificación final del proyecto en excelente, bueno, aceptable, cuestionable y malo.

Después de recibir los valores del peso relativo de cada criterio se construye la Tabla 1.

Tabla 1 Peso otorgado por los expertos a los criterios

G	C/E	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	Ep
25	C ₁	8	10	5	7	8	8	9	7.857142
	C ₂	10	8	15	12	10	10	10	10.71428
	C ₃	7	7	5	6	7	7	6	6.42857
30	C ₄	15	10	12	13	12	15	10	12.42857
	C ₅	6	10	10	10	8	9	10	9
	C ₆	9	10	8	7	10	6	10	8.57142
25	C ₇	9	10	12	8	10	9	9	9.57142
	C ₈	9	7	8	10	8	9	8	8.42857
	C ₉	7	8	5	7	7	7	8	7
20	C ₁₀	10	10	10	10	10	10	10	10
	C ₁₁	10	10	10	10	10	10	10	10
T		100	100	100	100	100	100	100	100

5. Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (χ^2). Siguiendo el procedimiento que se muestra a continuación.

- Sea C el número de criterios que van a evaluarse y E el número de expertos que realizan la evaluación.
- Para cada criterio se determina la $\sum E$ que representa la sumatoria del peso dado por cada experto, Ep que es la puntuación promedio de los pesos correspondientes a cada criterio, $M\sum E$ es la media de la $\sum E$ y ΔC que se obtiene calculando la diferencia entre $\sum E$ y $M\sum E$.

- Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión S por la expresión:

$$S = \sum(\sum E - \sum \sum E/C)^2$$

Tabla 2 Cálculo de la Dispersión (S) para hallar la concordancia entre los expertos

	$\sum E$	$\sum E/C$	$\sum E - \sum \sum E/C$	$(\sum E - \sum \sum E/C)^2$
C₁	55	5	-8.6363	74.5866
C₂	75	6.8181	11.36	129.1427
C₃	45	4.0909	-18.63	347.2967
C₄	87	7.9090	23.36	545.8811
C₅	63	5.7272	-0.63	0.4043
C₆	60	5.4545	-3.63	13.2197
C₇	67	6.0909	3.36	11.3171
C₈	59	5.3636	-4.63	21.4915
C₉	49	4.4545	-14.63	214.2095
C₁₀	70	6.3636	6.36	40.5017
C₁₁	70	6.3636	6.36	40.501768
$\sum \sum E/C$	-	63.6363	-	-
$S = \sum(\sum E - \sum \sum E/C)^2$	-	-	-	1438.553

- Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall W:

$$W = S/E^2(C^3 - C)/12$$

- El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real:

$$X^2 = E * (C - 1) * W$$

Los valores obtenidos se muestran en la tabla 3.

Tabla 3 Tabla para el cálculo de concordancia

Expertos/Criterios	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	ΣE	E _p	ΔC	ΔC ²
C₁	8	10	5	7	8	8	9	55	7.85714	-8	64
C₂	10	8	15	12	10	10	10	75	10.7142	12	144
C₃	7	7	5	6	7	7	6	45	6.4285	-18	324
C₄	15	10	12	13	12	15	10	87	12.4285	24	576
C₅	6	10	10	10	8	9	10	63	9	0	0
C₆	9	10	8	7	10	6	10	60	8.5714	-3	9
C₇	9	10	12	8	10	9	9	67	9.5714	4	16
C₈	9	7	8	10	8	9	8	59	8.4285	-4	16
C₉	7	8	5	7	7	7	8	49	7	-14	196
C₁₀	10	10	10	10	10	10	10	70	10	7	49
C₁₁	10	10	10	10	10	10	10	70	10	7	49
DC	100	100	100	100	100	100	100	100	90.4285	7	1443
M ΣE	63										
W	0.34938										
X²	18.6825										
X²_(α, c-1)	23.20										

- El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas, si se cumple que el $X_{\text{real}}^2 < X_{(\alpha, c-1)}^2$ se puede decir que existe concordancia en el trabajo de los expertos.

En el presente caso se puede afirmar que existe concordancia en el trabajo de los expertos:

$$18.682511 < 23.2093$$

Si no existe concordancia se hace necesario repetir el trabajo de los expertos.

6. Posteriormente se identifica el peso relativo de cada criterio P y se calcula el Índice de Aceptación (IA) de la propuesta.

Para esto se utiliza el procedimiento siguiente (Ver Tabla 4).

Conociendo el número de experto que realizan la evaluación E y la sumatoria de las puntuaciones de cada criterio C se puede calcular el peso de cada criterio P.

Conociendo el peso de cada criterio P y la calificación dada por los evaluadores c en una escala de 1 a 5 que se recogieron en el Modelo 2 (Ver Anexo 6) se puede calcular el valor de $P \times c$. Con el valor anterior se calcula el Índice de Aceptación del proyecto (IA). $IA = P * c/5$

Tabla 4 Calificación de cada criterio

Criterios	Calificación (c)					P	P x c
	1	2	3	4	5		
C1					X	0.07857	0.3928
C2					X	0.1071	0.535
C3					X	0.06428	0.32142
C4					X	0.1242	0.6214
C5					X	0.09	0.45
C6				X		0.0857	0.3428

C7					X	0.09571	0.47857
C8				X		0.084285	0.337142
C9			X			0.07	0.21
C10					X	0.10	0.5
C11					X	0.10	0.5
Total							4.6899
IA	0.938						

7. Por último se determina la probabilidad de éxito de la propuesta, ubicando el IA calculado anteriormente en rangos que están predefinidos (Ver tabla 5), en dependencia de donde se ubique será la probabilidad de éxito que tenga la propuesta.

Tabla 5 Rangos predefinidos de Índice de Aceptación

$0.7 < IA$	Existe alta probabilidad de éxito
$0.5 < IA < 0.7$	Existe probabilidad media de éxito
$0.3 < IA < 0.5$	Probabilidad de éxito baja
$IA < 0.3$	Fracaso seguro

El IA calculado es 0.938 lo que significa que existe alta probabilidad de éxito.

3.3 Conclusiones

Para la validación técnica de la propuesta se utilizó el método Experto. Con él se alcanzaron resultados favorables, obteniéndose una probabilidad de éxito alta que demuestra que lo planteado hasta el momento se adapta valiosamente a las condiciones existentes y aporta novedosos elementos que resultan imprescindibles para el desarrollo del SWE.

CONCLUSIONES GENERALES

Con el desarrollo de este trabajo de investigación, se obtuvieron las siguientes conclusiones:

- Se realizó el análisis de las características y elementos estructurales del SWE producido en la UCI para el MINED, con el objetivo de llegar a una solución propuesta.
- Se realizó una propuesta de los elementos arquitectónicos del software educativo para su futura utilización en los proyectos productivos en la UCI, para el MINED.
- Se creó un documento donde se recogió todo el proceso investigativo de la búsqueda de los elementos arquitectónicamente significativos del SWE que se produce en la UCI para el MINED.
- Luego de realizar la evaluación técnica de la propuesta, mediante el método Experto, se obtuvo una alta probabilidad de éxito, lo que implica desde el punto de vista teórico, el cumplimiento de la idea a defender planteada.

RECOMENDACIONES

Para perfeccionar el contenido del presente trabajo se recomienda:

- Aplicar la propuesta de solución en el proceso de desarrollo del software educativo del MINED que se producen en nuestra universidad, para probar su veracidad y efectividad.
- Extender la solución a otros proyectos productivos de la universidad, aunque no desarrollen SWE.

REFERENCIAS BIBLIOGRÁFICAS

Barrera Yanes, Rafael. 1998. *Del objetivo al guión interactivo*. La Habana : s.n., 1998. Vol. Vol 1.

Cataldi, Zulma. 2000. Metodología de diseño, desarrollo y evaluación de software educativo. [Online] 2000. Tesis de Magíster en Informática. (versión resumida).Facultad de Informática. <http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf>.

Fajardo, Elena Galán. El guión didáctico para materiales multimedia. [Online] [Cited: 2 4, 2008.] Universidad Carlos III de Madrid. <http://www.ucm.es/info/especulo/numero34/guionmu.html>.

Gregorio R.G., Javier G.F.and Eduardo G.J. 2008. metodología de investigación cualitativa. [Online] 2 6, 2008. <http://www.lapaginadelprofe.cl/guialesis/31cualitativa.htm>.

Huertas, Pérez. 1998. Introducción a la multimedia: realización y producción de programas. [Online] 1998. [Cited: 3 4, 2008.] Madrid: IORTV.

John C. Georgas, Eric M. Dashofy, y Richard N. Taylor. Desarrollo Centrado en la Arquitectura: Un acercamiento diferente a la Ingeniería de Software. [Online] [Cited: 2 4, 2008.] <http://www.acm.org/crossroads/espanol/xrds12-4/arqcentric.html>.

Kruchten, Philippe. 1995. *The 4+1 View Model of Architecture*. 1995. Vol. vol. 12.

Lara, Luis R. Introducción a un modelo complejo de los softwares multimediales educativos. [Online] [Cited: 3 5, 2008.] Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de Catamarca. Proyecto de Investigación PROTEO. <http://www.um.es/ead/red/12/lara.pdf>.

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 2da Edición. s.l. : Prentice Hall Hispanoamericana, 1999.

Manuel Gértrudix Barrio, Sergio Álvarez García, Antonio Galisteo del Valle, María del Carmen Gálvez de la Cuesta, Felipe Gértrudix Barrio. 2007. Acciones de diseño y desarrollo de objetos educativos digitales: programas institucionales. [Online] abril 2007. artículo. http://www.uoc.edu/rusc/4/1/dt/esp/gertrudix_alvarez_galisteo_galvez.pdf.

Marqués, Pere. 1996. El software educativo. [Online] 1996. http://www.lmi.ub.es/te/any96/marques_software/.

MORA, A. J. H. Multimedia. [Online] [Cited: 4 15, 2008.] <http://www.monografias.com/trabajos7/mult/mult2.shtml#his>.

Nurleidis Almeida Cintra and Yoannia Viera Cisneros. 2007. *Principios para la evaluación y certificación de la calidad de los productos de Software Educativo en la Universidad de las Ciencias Informáticas*. Ciudad de la Habana : s.n., 2007. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

Pérez Huertas, F. J. 1998. *Introducción a la multimedia: realización y producción de programas*. 1998. (Unidad didáctica 158).

Pressman, Roger S. 2002. *Ingeniería de Software. Un enfoque práctico (Traducción del original en inglés: Software Engineering A practical Approach)*. 5ta edición. Mac Graw Hill : s.n., 2002.

Ricardo, Febe Angel Ciudad. 2006. II TALLER DE SOFTWARE EDUCATIVO E HIPERMEDIA. [Online] Abril 2006. Ciudad de la Habana, Cuba. http://www.informaticahabana.com/evento_virtual/files/MUL053.pdf.

Ricardo, Febe Angel Ciudad. 2006. PRESENTACIÓN DEL LENGUAJE DE MODELACIÓN PARA APLICACIONES EDUCATIVAS ApEM – L 1.0. [Online] 2006. Ciudad de La Habana. http://virtualst.fordes.co.cu/EVirtual/files/Trabajo_23.doc.

Ricardo, Febe Angel Ciudad. 2006. Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. [Online] [Cited: 5 4, 2008.] <http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml>.

Río, Agustín Cernuda. 2002. Sistema de verificación de componentes software. [Online] Febrero 2002. TESIS DOCTORAL. Departamento de Informática. UNIVERSIDAD DE OVIEDO. <http://willydev.net/.../2008.05.07.Tesis.sistema%20de%20verificacion%20de%20componentes%20software-tesis.pdf>.

Rodríguez Liván A. SOFTWARE EDUCATIVO. HACIA UNA NUEVA PEDAGOGÍA BASADA EN LAS TIC. [Online] [Cited: 4 25, 2008.] <http://rvarela.ispvc.rimed.cu/articulos/rv1801.pdf>.

Shaw, M. y Garlan, D. 1996. *Software Architecture. Perspectives of an Emerging Discipline*, Prentice Hall. 1996.

2008. Software. [Online] 4 21, 2008. <http://es.wikipedia.org/wiki/Software>.

Standish. 1994. The CHAOS Report. [Online] 1994. http://www.standishgroup.com/sample_research/chaos.

SUÁREZ, D. A. D. V. M. ANTECEDENTES PEDAGÓGICOS DEL USO DE LA TECNOLOGÍA MULTIMEDIA EN LA EDUCACIÓN. [Online] [Cited: 5 20, 2008.] <http://www.servicio.cid.uc.edu.ve/educacion/revista/a5n26/5-26-10.pdf>.

Spiegel M. R 1977. *Teoría y problemas de estadística*. 1977. Volume, DOI:.

UNESCO. 2001. Ediciones UNESCO. Santillana : s.n., 2001.

VAQUERO, A. 1997. «Las TIC para la enseñanza, la formación y el aprendizaje». 1997. En Revista Novatita 132: Monografía sobre las TIC en la Educación.

Wolf and Perry. 1992. *Foundations for the Study of Software Architecture*. 1992. ACM Software Engineering Notes, 17, 4,.

ZULMA CATALDI, FERNANDO LAGE, RAÚL PESSACQ, RAMÓN GARCÍA-MARTÍNEZ. METODOLOGÍA EXTENDIDA PARA LA CREACIÓN DE SOFTWARE EDUCATIVO DESDE UNA VISIÓN INTEGRADORA. [Online] [Cited: 5 6, 2008.] Buenos Aires, Argentina. Volumen 2. REVISTA LATINOAMERICANA DE TECNOLOGÍA EDUCATIVA. <http://www.itba.edu.ar/capis/webcapis/RGMITBA/articulosrgm/R-extremadura-2.pdf>.

2008. La Historia del Software. [Online] 3 27, 2008. <http://cellular.ci.ulsu.mx/comun/historiaw/node23.html>.

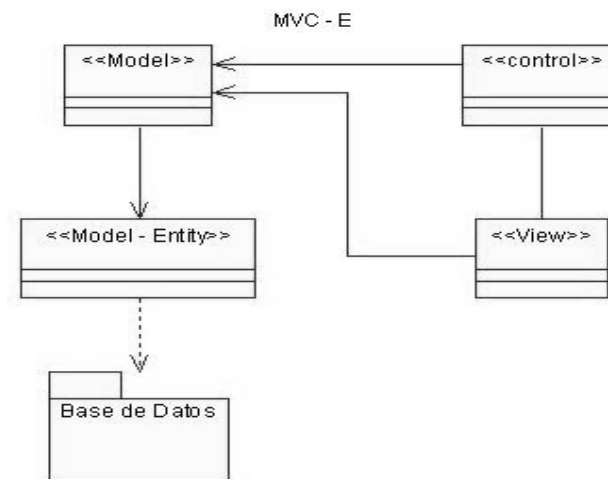
2008. Software. [Online] 4 21, 2008. <http://es.wikipedia.org/wiki/Software>.

BIBLIOGRAFÍA

- Barrera Yanes, Rafael. 1998.** *Del objetivo al guión interactivo*. La Habana : s.n., 1998. Vol. Vol 1.
- Cataldi, Zulma. 2000.** Metodología de diseño, desarrollo y evaluación de software educativo. [Online] 2000. Tesis de Magíster en Informática. (versión resumida).Facultad de Informática. <http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf>.
- Fajardo, Elena Galán.** El guión didáctico para materiales multimedia. [Online] [Cited: 2 4, 2008.] Universidad Carlos III de Madrid. <http://www.ucm.es/info/especulo/numero34/guionmu.html>.
- Gregorio R.G., Javier G.F.and Eduardo G.J. 2008.** metodología de investigación cualitativa. [Online] 2 6, 2008. <http://www.lapaginadelprofe.cl/guiatesis/31icualitativa.htm>.
- Huertas, Pérez. 1998.** Introducción a la multimedia: realización y producción de programas. [Online] 1998. [Cited: 3 4, 2008.] Madrid: IORTV.
- John C. Georgas, Eric M. Dashofy, y Richard N. Taylor.** Desarrollo Centrado en la Arquitectura: Un acercamiento diferente a la Ingeniería de Software. [Online] [Cited: 2 4, 2008.] <http://www.acm.org/crossroads/espanol/xrds12-4/arqcentric.html>.
- Kruchten, Philippe. 1995.** *The 4+1 View Model of Architecture*. 1995. Vol. vol. 12.
- Lara, Luis R.** Introducción a un modelo complejo de los softwares multimediales educativos. [Online] [Cited: 3 5, 2008.] Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de Catamarca.Proyecto de Investigación PROTEO. <http://www.um.es/ead/red/12/lara.pdf>.
- Larman, Craig. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 2da Edición. s.l. : Prentice Hall Hispanoamericana, 1999.
- Manuel Gértrudix Barrio, Sergio Álvarez García, Antonio Galisteo del Valle, María del Carmen Gálvez de la Cuesta, Felipe Gértrudix Barrio. 2007.** Acciones de diseño y desarrollo de objetos educativos digitales: programas institucionales. [Online] abril 2007. artículo. http://www.uoc.edu/rusc/4/1/dt/esp/gertrudix_alvarez_galisteo_galvez.pdf.
- Marqués, Pere. 1996.** El software educativo. [Online] 1996. http://www.lmi.ub.es/te/any96/marques_software/.
- MORA, A. J. H.** Multimedia. [Online] [Cited: 4 15, 2008.] <http://www.monografias.com/trabajos7/mult/mult2.shtml#his>.
- Nurileidis Almeida Cintra and Yoannia Viera Cisneros. 2007.** *Principios para la evaluación y certificación de la calidad de los productos de Software Educativo en la Universidad de las Ciencias Informáticas*. Ciudad de la Habana : s.n., 2007. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.
- Pérez Huertas, F. J. 1998.** *Introducción a la multimedia: realización y producción de programas*. 1998. (Unidad didáctica 158).
- Pressman, Roger S. 2002.** *Ingeniería de Software. Un enfoque práctico (Traducción del original en inglés: Software Engineering A practical Aproach)*. 5ta edición. Mac Graw Hill : s.n., 2002.
- Ricardo, Febe Angel Ciudad. 2006.** II TALLER DE SOFTWARE EDUCATIVO E HIPERMEDIA. [Online] Abril 2006. Ciudad de la Habana, Cuba. http://www.informaticahabana.com/evento_virtual/files/MUL053.pdf.

- Ricardo, Febe Angel Ciudad. 2006.** PRESENTACIÓN DEL LENGUAJE DE MODELACIÓN PARA APLICACIONES EDUCATIVAS ApEM – L 1.0. [Online] 2006. Ciudad de La Habana. http://virtualst.fordes.co.cu/EVirtual/files/Trabajo_23.doc.
- Ricardo, Febe Angel Ciudad. 2006.** Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos. [Online] [Cited: 5 4, 2008.] <http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml>.
- Río, Agustín Cernuda. 2002.** Sistema de verificación de componentes software. [Online] Febrero 2002. TESIS DOCTORAL. Departamento de Informática. UNIVERSIDAD DE OVIEDO. <http://willydev.net/.../2008.05.07.Tesis.sistema%20de%20verificacion%20de%20componentes%20software-tesis.pdf>.
- Rodríguez Liván A.** SOFTWARE EDUCATIVO. HACIA UNA NUEVA PEDAGOGÍA BASADA EN LAS TIC. [Online] [Cited: 4 25, 2008.] <http://rvarela.ispvc.rimed.cu/articulos/rv1801.pdf>.
- Shaw, M. y Garlan, D. 1996.** *Software Architecture. Perspectives of an Emerging Discipline*, Prentice Hall. 1996.
- 2008.** Software. [Online] 4 21, 2008. <http://es.wikipedia.org/wiki/Software>.
- Standish. 1994.** The CHAOS Report. [Online] 1994. http://www.standishgroup.com/sample_research/chaos.
- SUÁREZ, D. A. D. V. M.** ANTECEDENTES PEDAGÓGICOS DEL USO DE LA TECNOLOGÍA MULTIMEDIA EN LA EDUCACIÓN. [Online] [Cited: 5 20, 2008.] <http://www.servicio.cid.uc.edu.ve/educacion/revista/a5n26/5-26-10.pdf>.
- Spiegel M. R 1977.** *Teoría y problemas de estadística*. 1977. Volume, DOI:.
- UNESCO. 2001.** Ediciones UNESCO. Santillana : s.n., 2001.
- VAQUERO, A 1997.** «Las TIC para la enseñanza, la formación y el aprendizaje». 1997. En Revista Novatita 132: Monografía sobre las TIC en la Educación.
- Wolf and Perry. 1992.** *Foundations for the Study of Software Architecture*. 1992. ACM Software Engineering Notes, 17, 4,.
- ZULMA CATALDI, FERNANDO LAGE, RAÚL PESSACQ, RAMÓN GARCÍA-MARTÍNEZ.** METODOLOGÍA EXTENDIDA PARA LA CREACIÓN DE SOFTWARE EDUCATIVO DESDE UNA VISIÓN INTEGRADORA. [Online] [Cited: 5 6, 2008.] Buenos Aires, Argentina. Volumen 2. REVISTA LATINOAMERICANA DE TECNOLOGÍA EDUCATIVA. <http://www.itba.edu.ar/capis/webcapis/RGMITBA/articulosrgm/R-extremadura-2.pdf>.
- 2008.** La Historia del Software. [Online] 3 27, 2008. <http://cellular.ci.ulsu.mx/comun/historiaw/node23.html>.
- 2008.** Software. [Online] 4 21, 2008. <http://es.wikipedia.org/wiki/Software>

Anexo 1: Esquema del MVC – E.



Anexo2: Estructura básica de los programas didácticos

El entorno de comunicación o interfaces

La interfaz es el entorno a través del cual los programas establecen el diálogo con sus usuarios, y es la que posibilita la interactividad característica de estos materiales. Está integrada por dos sistemas:

- El sistema de comunicación programa-usuario, facilita la transmisión de informaciones al usuario por parte del ordenador que incluye:
 - Las pantallas a través de las cuales los programas presentan información a los usuarios.
 - Los informes y las fichas que se proporcionan mediante las impresoras.
 - El empleo de otros periféricos: altavoces, sintetizadores de voz, robots, módems, convertidores digitales-analógicos.
- El sistema de comunicación usuario-programa, proporciona la transmisión de información del usuario hacia el ordenador que incluye:
 - El uso del teclado y el ratón, mediante los cuales los usuarios introducen al ordenador un conjunto de órdenes o respuestas que los programas reconocen.

- El empleo de otros periféricos: micrófonos, lectores de fichas, teclados conceptuales, pantallas táctiles, lápices ópticos, módems, lectores de tarjetas, convertidores analógico-digitales entre otros.

Con la ayuda de las técnicas de la Inteligencia Artificial y del desarrollo de las tecnologías multimedia, se investiga la elaboración de entornos de comunicación cada vez más intuitivos y capaces de proporcionar un diálogo abierto y próximo al lenguaje natural.

Las bases de datos

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

- **Modelos de comportamiento**, representan la dinámica de un sistema. Se distinguen:
 - **Modelos físico-matemáticos**, que tienen leyes perfectamente determinadas por ecuaciones.
 - **Modelos no deterministas**, regidos por leyes no totalmente deterministas, que son representadas por ecuaciones con variables aleatorias, por grafos y por tablas de comportamiento.
- **Datos de tipo texto**, información alfanumérica.
- **Datos gráficos**, las bases de datos pueden estar constituidas por dibujos, fotografías y secuencias de video.
- **Sonido**, como los programas que permiten componer música, escuchar determinadas composiciones musicales y visionar sus partituras.

El motor o algoritmo

El algoritmo del programa, en función de las acciones de los usuarios, gestiona las secuencias en que se presenta la información de las bases de datos y las actividades que pueden realizar los alumnos. Se distinguen 4 tipos de algoritmos:

- **Lineal**, cuando la secuencia de las actividades es única.
- **Ramificado**, cuando están predeterminadas las posibles secuencias según las respuestas de los alumnos.
- **Tipo de entorno**, cuando no hay secuencias predeterminadas para el acceso del usuario a la información principal y a las diferentes actividades. El estudiante elige qué hacer y cuándo lo va a hacer. Este entorno puede ser:

- **Estático**, si el usuario sólo puede consultar (y en algunos casos aumentar o disminuir) la información que proporciona el entorno, pero no puede modificar su estructura.
 - **Dinámico**, si el usuario, además de consultar la información, también puede modificar el estado de los elementos que configuran el entorno.
 - **Programable**, si a partir de una serie de elementos el usuario puede construir diversos entornos.
 - **Instrumental**, si ofrece a los usuarios diversos instrumentos para realizar determinados trabajos.
- **Tipo de sistema de experto**, cuando el programa tiene un motor de inferencias y, mediante un diálogo bastante inteligente y libre con el alumno (sistemas dialógicos), asesora al estudiante o tutoriza inteligentemente el aprendizaje. Su desarrollo está muy ligado con los avances en el campo de la Inteligencia Artificial.

Anexo 3: Características del proceso de enseñanza-aprendizaje cubano

El problema de la comunicación en el proceso de enseñanza aprendizaje tiene gran actualidad, por las propias necesidades que está teniendo el sistema educativo actual. Variados son los enfoques que tratan de buscar una explicación a tan intrincado problema, el cual puede ser conceptualizado desde diferentes ciencias, al ser concebido desde el paradigma de la complejidad.

Actualmente nuestro país está enfrascado en una revolución educacional en todos los niveles, muestra de ello es la universalización de la enseñanza superior, por lo que se hace necesario la utilización de otro conjunto de medios que apoyen estos empeños, garanticen una transmisión correcta de los contenidos a impartir y guíen el proceso de enseñanza-aprendizaje en cualquier lugar de la nación, hacia los mismos objetivos de formación. De esta forma se garantiza que los egresados respondan al encargo social para el cual se formaron.

En los momentos actuales donde se pretende formar un joven que posea una cultura general integral, se hace necesario que se busquen vías que permitan contribuir con este fin, a partir de los diferentes enfoques didácticos existentes para la enseñanza. Dentro de las vías que pudieran resultar eficientes en el proceso de enseñanza aprendizaje, lo constituye el uso de las llamadas tareas integradoras. Estas basan su esencia en orientar al

estudiante a la resolución de una actividad o situación problemática, vinculada con la vida cotidiana o la práctica escolar a través de la computadora.

Algunos métodos de enseñanza.

Método deductivo

Cuando el asunto estudiado procede de lo general a lo particular. El profesor presenta conceptos, principios, definiciones o afirmaciones de las que se van extrayendo conclusiones y consecuencias, o se examinan casos particulares sobre la base de las afirmaciones generales presentadas. Si se parte de un principio, por ejemplo el de Arquímedes, en primer lugar se enuncia el principio y posteriormente se enumeran o exponen ejemplos de flotación. Los métodos deductivos son los que tradicionalmente más se utilizan en la enseñanza. Sin embargo, no se debe olvidar que para el aprendizaje de estrategias cognoscitivas, creación o síntesis conceptual, son los menos adecuados. Recordemos que en el aprendizaje propuesto desde el comienzo de este texto, se aboga por métodos experimentales y participativos.

El método deductivo es muy válido cuando los conceptos, definiciones, fórmulas, leyes y principios ya están muy asimilados por el alumno, pues a partir de ellos se generan las 'deducciones'. Evita trabajo y ahorra tiempo.

Método inductivo

Cuando el asunto estudiado se presenta por medio de casos particulares, sugiriéndose que se descubra el principio general que los rige. Es el método, activo por excelencia, que ha dado lugar a la mayoría de descubrimientos científicos. Se basa en la experiencia, en la participación, en los hechos y posibilita en gran medida la generalización y un razonamiento globalizado.

El método inductivo es el ideal para lograr principios, y a partir de ellos utilizar el método deductivo. Normalmente en las aulas se hace al revés. Si seguimos con el ejemplo iniciado más arriba del principio de Arquímedes, en este caso, de los ejemplos pasamos a la 'inducción' del principio, es decir, de lo particular a lo general. De hecho, fue la forma de razonar de Arquímedes cuando descubrió su principio.

Heurístico o de descubrimiento (del griego heurisko: enseñar)

Consiste en que el profesor incite al alumno a comprender antes de fijar, implicando justificaciones o fundamentaciones lógicas y teóricas que pueden ser presentadas por el profesor o investigadas por el alumno.

En el **aprendizaje significativo** interactúan dos factores: uno se relaciona con el material y el otro con la estructura cognoscitiva del estudiante, el cual es considerado como el factor principal que influye en el aprendizaje y garantiza la retención significativa de los nuevos conocimientos.

El aprendizaje significativo, para Ausubel, tiene lugar cuando las nuevas informaciones y conocimientos, pueden relacionarse de una manera no arbitraria con lo que la persona ya sabe. Por este motivo, los conocimientos previos tienen relevancia en la medida en que la asociación entre ellos y los nuevos conceptos se establecen. De allí que la motivación para el aprendizaje se da en la medida en que está comprometido el aspecto afectivo, por la satisfacción que de forma espontánea aparece. Cuando esto no se da, se produce un elevado estado de ansiedad y experiencias de fracaso, que llevan a la alternativa de aprendizaje memorístico.

Anexo 4: Entrevista

Fecha de envío:

Fecha de respuesta:

Nombre del proyecto: _____

Nombre del producto: _____

Nombre y apellidos del entrevistado:

Cargo que ocupa en el proyecto:

Objetivo del proyecto:

Objetivo del producto:

Breve descripción del producto:

Preguntas:

1. ¿Se utiliza algún patrón arquitectónico en el proceso de desarrollo del SWE para el MINED?

_____ si _____ no

2. En caso de utilizarse cual o cuales utilizan.

3. ¿Qué tipología de SWE se producen en el MINED? Marque con una x
- Ejercitación Tutorial Base de Datos
- Libro Simulador Juego
- Constructor Herramientas
4. ¿Que funcionalidades debe cumplir el producto SWE para el MINED?
6. ¿Ves en el patrón MVC_E un mayor acercamiento al SWE cubano?
- Si No
7. En caso de responder si diga .Explique porque?
8. Marque con una x las funciones que realiza su producto de software educativo.
- Se hacen preguntas a los estudiantes.
 - El ordenador adopta el papel de juez poseedor de la verdad y examina al alumno.
 - Se producen errores cuando la respuesta del alumno está en desacuerdo con la que el ordenador tiene como correcta.
 - El ordenador no juzga las acciones del alumno, se limita a procesar los datos que éste introduce y a mostrar las consecuencias de sus acciones sobre un entorno
 - El software proporciona un esqueleto, una estructura, sobre la cual los alumnos y los profesores pueden añadir el contenido que les interese.
 - Estructuran los contenidos en niveles de dificultad.
 - Proporciona a los alumnos una serie de herramientas de búsqueda y de proceso de la información que pueden utilizar libremente para construir la respuesta a las preguntas del programa.
 - Proporciona datos organizados, en un entorno estático, según determinados criterios, y facilita su exploración y consulta selectiva.
 - Acostumbran a realizar preguntas del tipo: ¿Qué características tiene este dato? ¿Qué datos hay con la característica X? ¿Qué datos hay con las características X y Y?
 - Los estudiantes pueden tomar decisiones y adquirir experiencias directas delante de situaciones que frecuentemente resultarían difícilmente accesibles a la realidad (control de una central nuclear, contracción del tiempo, pilotaje de un avión)

- Facilita el desarrollo de los reflejos y la percepción visual.
- La investigación de los estudiantes/experimentadores puede realizarse en tiempo real o en tiempo acelerado.
- Acostumbran a realizar preguntas del tipo: ¿Qué pasa al modelo si modifico el valor de la variable X? ¿Y si modifico el parámetro Y?
- Facilita a los usuarios elementos simples con los cuales pueden construir elementos más complejos o entornos.
- Acostumbran a realizar pregunta del tipo: ¿Qué sucede si añado o elimino el elemento X?

Anexo 5: Modelo 1 de la encuesta para la validación técnica de la propuesta.

Guía para informar el peso de los criterios.

Fecha de recepción _____

Fecha de entrega _____

Nombre y Apellidos del evaluador _____

Pesos Relativos

Grupo No. 1.....25

Grupo No. 2.....30

Grupo No.3.....25

Grupo No.4.....20

Usted como experto debe conceder pesos a cada uno de los criterios establecidos, teniendo en cuenta que la suma de los valores dados, para un grupo, no exceda del peso relativo asignado a este.

Grupo No 1: Criterios de mérito científico.

1. Valor científico de la propuesta.

Peso ___

2. Calidad de la investigación.

Peso ___

3. Contribución científica.

Peso __

Grupo No 2: Criterios de implantación

4. Necesidad de empleo de la propuesta.

Peso __

5. Obtención de productos finales con calidad.

Peso __

6. Posibilidades de aplicación.

Peso __

Grupo No 3: Criterios de flexibilidad.

7. Adaptabilidad a proyectos productivos de SWE independientemente del tipo de SWE que desarrollen.

Peso __

8. Capacidad de la solución propuesta para adecuarse al proceso de desarrollo del SWE producido para el MINED.

Peso __

9. Capacidad de la solución propuesta para adecuarse a los procesos de desarrollo de proyectos productivos que no desarrollen SWE.

Peso __

Grupo No 4. Criterios de impacto.

10. Efectos en la mejora del proceso de desarrollo del SWE para el MINED.

Peso __

11. Repercusión en la calidad de los procesos de desarrollo en los proyectos productivos de SWE para el MINED.

Peso __

Anexo 6: Modelo 2 de la encuesta para la validación técnica de la propuesta.

Guía para la evaluación.

Fecha de recepción _____

Fecha de entrega _____

Nombre y Apellidos del evaluador _____

Usted como experto debe conceder una evaluación cuantitativa de cada criterio con una escala de 1 a 5.

Grupo No 1: Criterios de mérito científico.

12. Valor científico de la propuesta.

Peso __

13. Calidad de la investigación.

Peso __

14. Contribución científica.

Peso __

Grupo No 2: Criterios de implantación

15. Necesidad de empleo de la propuesta.

Peso __

16. Obtención de productos finales con calidad.

Peso __

17. Posibilidades de aplicación.

Peso __

Grupo No 3: Criterios de flexibilidad.

18. Adaptabilidad a proyectos productivos de SWE independientemente del tipo de SWE que desarrollen.

Peso __

19. Capacidad de la solución propuesta para adecuarse al proceso de desarrollo del SWE producido para el MINED.

Peso __

20. Capacidad de la solución propuesta para adecuarse a los procesos de desarrollo de proyectos productivos que no desarrollen SWE.

Peso __

Grupo No 4. Criterios de impacto.

21. Efectos en la mejora del proceso de desarrollo del SWE para el MINED.

Peso __

22. Repercusión en la calidad de los procesos de desarrollo en los proyectos productivos para el MINED.

Peso __

Expresa su apreciación cualitativa con una clasificación final del proyecto en excelente, bueno, aceptable, cuestionable o malo.

___ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

___ Bueno: Novedad científica, resultados destacados.

___ Aceptable: Suficientemente bueno con reservas.

___ Cuestionable: No tiene relevancia científica y los resultados son malos.

___ Malo: No aplicable.

GLOSARIO DE TÉRMINOS Y SIGLAS

Aprendizaje cognoscitivo: Trata de explicar como los animales y el hombre pueden aprender conductas nuevas sin experiencia previa, o como se pueden recordar respuestas de gran complejidad durante un período largo de tiempo y sin reforzamiento, o como se pueden realizar aprendizajes de gran complejidad. Se considera al organismo un ser activo, capaz de elaborar la información y de generar conductas por motivaciones internas. Este aprendizaje subraya los aspectos cognitivos, se basa en representaciones cognitivas de la conducta, en vez de la asociación de estímulos y respuestas. Sólo se da en especies animales superiores y en el hombre. El aprendizaje se puede realizar no solo por condicionamiento, sino que podemos aprender imitando a otros sujetos o simplemente al recibir la información de algo. Se llama aprendizaje vicario, observacional o por modelos.

Aprendizaje significativo: El aprendizaje significativo es un aprendizaje relacional. El sentido lo da la relación del nuevo conocimiento con: conocimientos anteriores, situaciones cotidianas, la propia experiencia y situaciones reales.

Aprendizaje: El aprendizaje es uno de los procesos más importantes para la psicología científica actual, es un cambio casi permanente en el comportamiento del organismo, mediante el cual es posible modificar lo que se ha aprendido anteriormente. A diferencia de los animales, que nacen con instrucciones genéticas para la supervivencia, los humanos tenemos la capacidad de aprendizaje, la cual nos da más flexibilidad para adaptarnos al medio ambiente. Podemos aprender a resguardarnos de cambios climáticos y adaptarnos a cualquier ambiente, nuestra capacidad de aprendizaje nos permite afrontar cambios.

Cognitivo: Proceso exclusivamente intelectual que precede al aprendizaje, las capacidades cognitivas solo se aprecian en la acción, es decir primero se procesa información y después se analiza, se argumenta, se comprende y se producen nuevos enfoques. El desarrollo de lo cognitivo en el alumno debe ser el centro del proceso de enseñanza por parte del docente. Este término es utilizado por la psicología moderna, concediendo mayor importancia a los aspectos intelectuales que a los afectivos y emocionales, en este sentido se tiene un doble significado: primero, se refiere a una representación conceptual de los objetos y segundo, es la comprensión o explicación de los objetos.

Componente: Una parte física reemplazable de un sistema que empaqueta su implementación, y es conforme a un conjunto de interfaces a las que proporciona su realización.

Entidad: Existencia no necesariamente material o animada.

Hipertexto: Un hipertexto es un documento digital o no, que se puede leer de manera no secuencial. Un hipertexto tiene los siguientes elementos: secciones, enlaces o hipervínculos y anclajes.

Hipertexto: Un hipertexto es un documento digital o no, que se puede leer de manera no secuencial. Un hipertexto tiene los siguientes elementos: secciones, enlaces o hipervínculos y anclajes.

Ingeniería del software: Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

Módulo: Un módulo es un componente auto controlado de un sistema, el cual posee una interfaz bien definida hacia otros componentes.

Peso: nivel de importancia asignado a cada una de las actividades que se pretenden medir.

Proceso: Conjunto de actividades, realizadas en forma secuencial, que realiza una organización, para crear, producir y entregar productos, de tal manera que satisfagan las necesidades de sus clientes.

Producto: Resultado concreto, observable y medible que surge como consecuencia del proceso, proyecto o experiencia desarrollada.

Proyecto: Es un elemento organizativo a través del cual se gestiona el desarrollo de software, su resultado es un producto.

Sistema: Colección de unidades conectadas que se organiza para lograr un propósito. El sistema es el “modelo completo”.

Software Educativo (SWE.): Se define como un programa automatizado diseñado con una intencionalidad educativa para ser utilizado en el proceso de aprendizaje, utiliza procedimientos para que el estudiante aprenda, se fomenta el análisis de problemas, facilita el trabajo en grupo, provee soporte en actividades docentes y en el sentido más amplio, mejora las habilidades del pensamiento y la resolución de problemas.

SIGLAS

CNICE: Centro Nacional de Información y Comunicación Educativa (Organización de España)

HLL: Lenguaje de Alto Nivel (por sus siglas en inglés correspondientes a High Level Language)

IA: Inteligencia Artificial

ICSW: Industria Cubana del Software

IEEE: Corresponde a las siglas del Instituto de Ingenieros Eléctricos y Electrónicos. Es una asociación estadounidense dedicada a la estandarización internacional, sin fines de lucro, formada por profesionales de las nuevas tecnologías.

ISW: Ingeniería de Software

LOM: Learning Object Metadata

LTSC: Learning Technology Standards Committer

MINED: Ministerio de Educación

MVC: Modelo-Vista-Controlador

MVC_E: Modelo-Vista-Controlador- Entidad

OCL: Object Constraint Language

ODE: Open Dynamics Engine

OMMMA_L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia.

POO: Programación Orientada a Objetos

SWE: Software Educativo

UCI: Universidad de las Ciencias Informáticas

UML: Lenguaje Unificado de Modelado