

Año 2009



UNIVERSIDAD NACIONAL DE LA PLATA -
FACULTAD DE INFORMÁTICA -
Secretaría de Postgrado

PROPUESTA PARA LOGRAR ESPECIALIZACION EN TIAE
TÍTULO: METODOLOGÍAS DE DISEÑO USADAS EN INGENIERÍA
WEB, SU VINCULACIÓN CON LAS NTICS

Alumna: Ana Nieves del Valle Rodríguez

Director : Mg. Prof. Rodolfo Bertone.

A mi querida familia: Marcelo, Gonzalo, Sebastián (que pusieron en práctica sus artes culinarias en los años que me demandó la carrera), en especial a Laura que sé que me necesitaba y no podía dedicarle el tiempo suficiente.

A Teté, a Contij, que me apoyaron aún sin saberlo.

En memoria de Dardo: mi amigo y motor, sin él nunca hubiese empezado esta cruzada.

Solo puedo decir Gracias!

Ana

Índice de Contenidos

Introducción	5
Estructura del Trabajo	6
Capítulo 1 – Ingeniería de Software - Ingeniería Web	7
1.1. Introducción.....	7
1.2. Definiciones previas. De Tecnología, Ciencias e Ingeniería.....	7
1.2.1. Tecnología	7
1.2.2. Ciencia	8
1.2.3. Ingeniería.....	8
1.3. Software – Un poco de historia	9
1.3.1. Ingeniería de software	9
1.3.2. El surgimiento de la Web	11
1.3.3. Ingeniería Web como un nuevo paradigma.....	12
1.4. Ingeniería Web	13
1.5. Ingeniería Web Vs. Ingeniería de Software	13
1.6. La Ingeniería Web obtiene un espacio	13
1.6.1 Primeros pasos en Ingeniería Web	14
Capítulo 2 – Metodologías de la Iweb.....	16
2.1. Introducción.....	16
2.2. Proceso de la IWeb	16
2.3. Métodos de la Ingeniería Web	17
2.3.1. Ejemplo con el que se trabajará en este capítulo:	17
2.3.2. Hypertext Design Model (HDM).....	18
2.3.3. Relationship Management Methodology (RMM)	20
2.3.4. Object Oriented Hypermedia DesignMethod (OOHDM)	23
2.3.5. Método OO-H.....	27
2.3.6. Object Oriented Web Solution (OOWS).....	28
2.3.7. UML based Web Engineering (UWE).....	30
2.4. Comparativa de Modelos	33
2.4.1. Conclusión	34
Capítulo 3 – Areas involucradas en el diseño de aplicaciones Web	35
3.1. Introducción.....	35
3.2. Consideraciones previas - Planificación.....	35
3.2.1. Consideraciones de Roger Pressman	35
3.3. Calidad	37
3.3.1. Otras definiciones de calidad	37
3.3.2. Cómo se mide la calidad de una WebApp?.....	37
3.3.3. Modelo de Mc Call	38
3.3.4. Modelo de Olsina	38
3.4. Usabilidad	39
3.4.1. Beneficios de la Usabilidad.....	41
3.4.2. UEM [Usability Evaluation Methods]	41
3.4.3. Métricas de usabilidad	42
3.4.4. Métricas de evaluación de calidad.....	42
3.5. Arquitectura de la Información (AI).....	42
Capítulo 4 – Del diseño Instruccional.....	44
4.1. Introducción.....	44
4.2. Aprendizaje significativo.....	44
4.2.1. Aprendizaje significativo:	44
4.2.1.1. Ventajas del Aprendizaje Significativo	45

4.3. Del Diseño Instruccional	47
4.3.1. Teorías del Diseño Instruccional	47
<i>Teoría de Elaboración de Reigeluth:</i>	48
<i>ISD (Instructional System Design) ó SAT (System Approach to Training):</i>	48
<i>Otros Modelos Prescriptivos Cognitivistas de Diseño:</i>	48
4.3.2. Resumen de Modelos Instruccionales	49
4.3.3. Estrategias de Diseño Instruccional:	49
4.4. De las teorías de aprendizaje.....	49
4.5. De la Ingeniería Web y el Diseño Instruccional	50
4.5.1. Beneficios que aporta el Diseño Instruccional a e-Learning [elearnspace].	50
4.5.2. Método de Especificación Instruccional de Software Educativo (EISE)	50
Figura 12 – Estructura de la especificación Instruccional de software educativo (Extraído de “Diseño Instruccional aplicado al desarrollo de software educativo – E. J. Valdemar)	52
Capítulo 5 – Conclusiones	53
5.1. Estructura del Diseño Instruccional y de la Ingeniería de Software	53
5.2. Estructura del Diseño Instruccional y de la Ingeniería Web.....	54
5.2.1. Ejemplo de uso de la Ingeniería Web en el Diseño Instruccional	55
5.3. Consideraciones finales	57

Índice de Figuras

Figura 1 – Notación HDM	19
Figura 2 – Ejemplo en HDM.....	19
Figura 3 – Notación del método RMM.....	22
Figura 4 - RMDM	22
Figura 5 – OOHDM (extraído de “Developing Hypermedia Applications using OOHDM” Daniel Schwabe and Gustavo Rossi).....	24
Figura 6 – Modelo Conceptual	25
Figura 7 – Diagrama de Clases Navegacionales	26
Figura 8 – Diagrama de Contextos Navegacionales	27
Figura 9 – Método OOWS (Extraído de “Desarrollo Dirigido por Modelos de Aplicaciones Web que integran Datos y Funcionalidad a partir de Servicios Web” de Ricardo Rafael Quintero Meza).....	29
Figura 10 – Notación UWE	32
Figura 11 – Método UWE aplicado al ejemplo.....	32
Figura 12 – Estructura de la especificación Instruccional de software educativo (Extraído de “Diseño Instruccional aplicado al desarrollo de software educativo – E. J. Valdemar)	52
Figura 13: Etapas de la Ingeniería de Software coincidentes con las etapas del diseño instruccional.....	53
Figura 14: Pantalla de ITE para la asignatura Diseño en la Web.....	56
Figura 15: Elementos Multimediales.....	56
Figura 16: Pantalla que refiere a conocimientos previos.....	57
Figura 17: Pantalla de ejercitación	57

Índice de Tablas

Tabla 1 – Histórico de la Ingeniería Web	15
Tabla 2 – Comparativa de Modelos de diseño	34
Tabla 3: Tabla comparativa de las Etapas en el desarrollo de software y del Diseño Instruccional	54

Introducción

“Encuéntrame, descárgame y úsame. Déjate luego seducir y vuelve a mí una y otra vez. Yo por mi parte usaré mucho, mucho sentido común y trataré de sorprenderte cada día.”

Esta metáfora de Ricardo Baeza Yates en “Cinco claves para la Web” presenta el espíritu de Internet cual es: **Encontrar Información de Calidad y Actualizada**, aunque sabemos que La Red no garantiza que esa información sea la apropiada. *“La Web es a la vez un mundo de sorpresas y desilusiones, es como navegar en un mar lleno de basura, contaminado, encontrando sólo a veces corales y perlas”*. Este es un punto muy importante a tener en cuenta en el proceso educativo, sin dudas que los alumnos **deben** usar las nuevas tecnologías, son parte de su vida, es normal que se valgan de Internet para realizar sus tareas, es allí en donde el sistema educativo debe participar: guiando, aconsejando, enseñando, participando en esos saberes aún no estructurados.

El problema es que, muchas veces, los docentes debemos lidiar con aplicaciones lentas con funcionalidad deficiente que no han sido estudiadas ni han tenido en cuenta su diseño estructural.

Por tal motivo, es necesario que se utilice una metodología apropiada para el desarrollo de aplicaciones Web, esto es: métodos formales de diseño e implementación de manera sistemática y modular, más si se trata de sistemas a gran escala.

Para Jakob Nielsen, la Web tuvo su propio problema del año 2000, producto de su tamaño y rapidez de crecimiento. Para las estimaciones realizadas por Ricardo Baeza Yates y Cuauhtémoc Rivera Loaiza - en el artículo “Ubicuidad y Usabilidad en la Web” publicado en el año 2002 - *en el año 2003 habría 100 millones de sitios web, esto significaba que habría que diseñar tantos sitios como los que había en 2002 solo en 12 meses*. Siguiendo este artículo, en ese momento *había a lo más 200 mil profesionales en Interfaces de Usuario (IU) y diseño de Webs en el mundo*, esto indicaba que *cada uno de ellos debería diseñar aproximadamente 5 sitios por semana. Por supuesto que esto no podía ocurrir y por lo tanto se proponían tres posibles soluciones:*

- 1. que sea posible diseñar sitios razonables sin ser experto en IUs;*
- 2. capacitar más gente en buen diseño de Web; y*
- 3. vivir con sitios mal diseñados que son difíciles de usar.*

De acuerdo a Nielsen, la última opción es inaceptable, pues en ese caso la Web dejará de ser usable y por lo tanto menos popular. Sin embargo esto es lo que ocurre en la gran mayoría de los sitios.

En este contexto, es fundamental la presencia del sistema educativo como formador de recursos humanos y como transmisor de conocimiento, que debe ser auxiliado mediante **recursos académicos** que permitan adecuarlo a la realidad y sociabilizarlo.

La doctora Roxana Morduchowicz expresa: *No les tengamos miedo a los medios, a la tecnología; acompañemos a los chicos en sus consumos...Hoy niños y jóvenes se constituyen en la cultura audiovisual. Los jóvenes no acceden al conocimiento solamente en la escuela, sino que adquieren saberes fragmentados, disímiles y diversos. Estos saberes están propiciados muchas veces por los medios y por ende con una lógica en su mayoría comercial. Por ello, es necesario primero que la escuela incorpore los gustos de niños y jóvenes para reconocerlos en su dimensión identitaria compleja y diversa, y además, porque en la escuela, es en el lugar que se puede resignificar social y conjuntamente, las representaciones y sentidos que los medios asignan a las personas y a los acontecimientos. Incorporar los medios al aula, como objetos de estudio, propiciando un uso crítico, retomando los gustos y consumos de sus alumnos, es un paso fundamental para trabajar más complejamente los contenidos curriculares y actualizar los procesos de enseñanza aprendizaje en el contexto social y cultural actual. El fracaso escolar*

– entre otras causas - se vincularía con la imposibilidad de la escuela de dar entrada a los saberes previos de sus alumnos....

Este Trabajo Integrador tiene como objetivo ofrecer un marco teórico que vincule la ingeniería web con cuestiones del ámbito del diseño instruccional apoyado con las Nuevas Tecnologías de Información y Comunicación.

Estructura del Trabajo

Este trabajo está organizado en cuatro capítulos que se resumen a continuación:

- **Capítulo 1:** Muestra cómo la web se ha ido incorporando en todas las actividades cotidianas. Contempla definiciones como tecnología, ciencia e ingeniería, particularmente la ingeniería de software que da pie al nacimiento de esta nueva disciplina: la Ingeniería Web. Ofrece un panorama general de su inserción, de la obtención de su propio espacio y la necesidad del uso de métodos y técnicas en el desarrollo de WebApp.
- **Capítulo 2:** Revisa el estado del arte de los Métodos usados en Ingeniería Web. Aplica algunos métodos a un ejemplo simple y por último se construye una tabla comparativa de los métodos vistos en este capítulo.
- **Capítulo 3:** Hace un análisis de las áreas involucradas en el diseño de aplicaciones Web teniendo en cuenta las consideraciones de diferentes autores.
- **Capítulo 4:** Trata temas relacionados con el Diseño Instruccional, el aprendizaje significativo y cómo las aplicaciones hipermedia se van insertando, casi en silencio, en el mundo educativo. Se presenta el contexto relacionado a la interacción entre la Ingeniería Web y los aspectos del Diseño Instruccional. Se analizan los beneficios que aporta, se presenta y analiza el método EISE.
- **Capítulo 5:** Conclusiones, presentación de una aplicación.

Capítulo 1 – Ingeniería de Software - Ingeniería Web

1.1. Introducción

“Navegar en la Web” se ha convertido en una actividad cotidiana, es evidente que Internet se popularizó. En la web se pueden encontrar sitios desde los más simples constituidos por un conjunto de páginas HTML vinculadas, buscadores de información, aulas virtuales, comercio electrónico y hasta aquellos que permiten realizar transacciones bancarias con alta seguridad. Actualmente y con la ayuda de software específico, crear un sitio Web, es una tarea fácil, accesible a cualquier persona que posea un toque de audacia, buen gusto, y constancia. Todo lo que necesita está en Internet: programas completos, applets, scripts, etc. En el ciberespacio se ha formado una comunidad “de hecho” y en gran medida solidaria, que se comunica mediante foros, salas de chat o mails y logran...resolver problemas planteados!

En estos sitios, improvisados, no existe un plan, ni siquiera un diseño, no han sido estudiados, generalmente se “inventan” en la marcha y posiblemente solucionan las necesidades del cliente. El cliente ignora que su sistema ha sido emparchado, que muchas piezas fueron apiñadas a presión y que en cualquier momento se puede “desmoronar”.

Las empresas, muchas veces prefieren pagar a estos pseudo-programadores, o capacitar durante poco tiempo a su personal, que logra obtener resultados en corto plazo, aunque el mismo, no sea eficiente. En muchas ocasiones, la empresa corre el riesgo de fracasar, si el navegante detecta estas deficiencias ya sea porque no encuentra la información o porque la aplicación es lenta o sobrecargada.

El principal objetivo de éste trabajo, como se dijo en la Introducción, es ofrecer un marco teórico que vincule la ingeniería web con cuestiones del ámbito del diseño instruccional apoyado con las Nuevas Tecnologías de Información y Comunicación.

Teniendo en cuenta el objetivo planteado, se hará un estudio teórico para responder algunas preguntas que surgen a partir de la experiencia:

¿Por qué decir ingeniería?, ¿Se puede hablar de una nueva disciplina?, ¿Se puede decir que existe la Ingeniería Web?, ¿Cómo aplicar los métodos de la Ingeniería Web al diseño instruccional?, ¿En qué mejora la calidad educativa un sitio planteado en el marco de la Ingeniería Web?.

Para realizar un análisis respecto de la Ingeniería de Software y de la Ingeniería Web, ambas disciplinas relativamente nuevas, es preciso ir hacia las raíces, aclarar cómo hacen uso de la tecnología y cuál es la intervención de la ciencia, por lo que en este capítulo se hará un análisis de sus acepciones.

1.2. Definiciones previas. De Tecnología, Ciencias e Ingeniería

1.2.1. Tecnología

Para la Real Academia Española (RAE): “Tecnología es un Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico. Lenguaje propio de una ciencia o un arte”.

Las tecnologías han sido inventadas desde tiempos remotos en diferentes lugares del mundo, desde armas y herramientas de piedra con más de un millón de años hasta llegar a la escritura, que marca un hito ya que se inician: el período histórico y los procesos sistemáticos de transmisión de información y de análisis racional de las tecnologías, procesos cuya muy posterior culminación sería el surgimiento de las ciencias.

Las tecnologías han tenido un fuerte impacto en la actividad humana, y desde un punto de vista económico, son un medio indispensable para obtener renta.

Herbert Marshall McLuhan & B. R. Powers en “La aldea global en la vida y los medios de comunicación mundiales en el siglo XXI”, realizan el análisis del impacto de las tecnologías, y plantea cuatro preguntas a contestar sobre cada tecnología particular:

1. ¿Qué genera, crea o posibilita?
2. ¿Qué preserva o aumenta?
3. ¿Qué recupera o revaloriza?
4. ¿Qué reemplaza o deja obsoleto?

Estas preguntas ayudan a establecer cómo las tecnologías afectan la vida humana (positiva o negativamente), por medio de ellas, se puede medir desde el impacto ambiental, hasta el epistemológico estableciendo qué conocimientos nuevos abre o potencia.

1.2.2. Ciencia

Para la Real Academia Española (RAE), ciencia es un “Conjunto de conocimientos obtenidos mediante la observación y el razonamiento, sistemáticamente estructurados y de los que se deducen principios y leyes generales”.

La ciencia intenta explicar fenómenos inexplicables o novedosos usando la observación y el razonamiento en forma sistemática (método científico) y para explicar los resultados experimentales (generalmente) se crean modelos matemáticos.

La ciencia abarca: (a) el proceso (adquisición sistemática - método científico) mediante el cual se adquiere conocimiento, y (b) el cuerpo organizado de conocimiento obtenido a través de este proceso. *“Mientras los animales inferiores sólo están en el mundo, el hombre trata de entenderlo; y sobre la base de su inteligencia imperfecta pero perfectible, del mundo, el hombre intenta enseñorarse de él para hacerlo más confortable. En este proceso, construye un mundo artificial: ese creciente cuerpo de ideas llamado “ciencia”, que puede caracterizarse como conocimiento racional, sistemático, exacto, verificable y por consiguiente falible. Por medio de la investigación científica, el hombre ha alcanzado una reconstrucción conceptual del mundo que es cada vez más amplia, profunda y exacta”.*

*En su sentido más amplio...(la ciencia)...se emplea para referirse al conocimiento en cualquier campo, pero que suele aplicarse sobre todo a la organización del proceso experimental verificable.*ⁱ

*La ciencia puede caracterizarse como conocimiento racional, exacto y verificable. Por medio de la investigación científica, el hombre ha alcanzado una reconstrucción conceptual del mundo que es cada vez más amplia, profunda y exacta.*ⁱⁱ

La ciencia necesita de un método experimental para ser confirmada, desarrolla leyes haciendo uso de la tecnología para sus avances.

Las tecnologías hacen uso de la experimentación, la ciencia usa el método científico, aunque la experimentación es también usada por las ciencias.

1.2.3. Ingeniería

“La ingeniería es transformar las ideas en realidad”

Para la Real Academia Española (RAE): “Ingeniería es el estudio y aplicación, por especialistas, de las diversas ramas de la tecnología. Actividad profesional del ingeniero”.

Para la RAE ingeniero, ra. (De ingenio, máquina o artificio): “Persona que profesa la ingeniería o alguna de sus ramas. Hombre que discurre con ingenio las trazas y modos de conseguir o ejecutar algo”.

Por tanto, la ingeniería es el *“Estudio y aplicación de las diversas ramas de la tecnología, por alguien que discurre con ingenio las trazas y modos de ejecutar algo”.*

La ingeniería nace con el comienzo de la revolución industrial, y constituye uno de los pilares del desarrollo de las sociedades modernas.

La ingeniería es la profesión que aplica conocimientos y experiencias para que mediante diseños, modelos y técnicas se resuelvan problemas que afectan a la humanidad.

Otro concepto que define a la ingeniería es el arte de aplicar los conocimientos científicos a la invención, perfeccionamiento o utilización de la técnica en todas sus determinaciones.

1.3. Software – Un poco de historia

Aunque la palabra software fue mencionada por W. Tukey en 1957, la teoría básica fue propuesta por el matemático hindú Alan Turing en 1935 en su ensayo “los números computables”. Si bien en 1941, cuando se inventó el primer computador Z3, aún no se usaba el término software, debía existir “algo” que lo hiciera funcionar.

En la Segunda Guerra Mundial, se impulsaron grandes avances en el desarrollo de hardware y por ende en el software ya que se debía automatizar el proceso de gestión, se pueden mencionar las máquinas Colossus que fueron primitivos dispositivos calculadores usados por los británicos para leer las comunicaciones alemanas cifradas. Colossus fue uno de los primeros computadores digitales, comparaba dos flujos de datos, contando cada coincidencia basada en una función programable booleana. El mensaje cifrado se leía a gran velocidad a través de una cinta de papel. El otro flujo de datos era generado internamente, y era una simulación electrónica de la máquina de Lorenz (la SZ40/42) en varias combinaciones. Si el número de coincidencias para una combinación era superior a una cierta cantidad, la salida era escrita en una máquina de escribir eléctrica.

En 1854 George Boole publicó “Investigación sobre las leyes del pensamiento”, un libro que trataba por completo de la lógica simbólica y su álgebra. El gran descubrimiento de Boole fue aplicar una serie de símbolos a operaciones lógicas que podían manipularse según reglas fijas que producirían resultados lógicos. Lo interesante es que no observó la lógica como una rama de las matemáticas, sino apuntaba una profunda relación entre el álgebra simbólica y lo que podía hacer para representar formas lógicas y silogismos, que nos podían ayudar a decir que las matemáticas estaban restringidas a dos cantidades: el 0 y el 1. Este trabajo pasó un siglo sin aplicación práctica. En 1937, Claude Shannon demuestra que las operaciones básicas del álgebra de Boole podían ser representadas mediante circuitos eléctricos, y que la combinación de varios circuitos podrían representar a operaciones lógicas complejas. Esto da inicio a la microelectrónica, y con ella los computadores de escritorio que dieron impulso a la industria del software, se comienzan a realizar aplicaciones para negocios, empresas, etc.

Esta explosión de necesidades de información y su organización, dio lugar a una nueva etapa a la que se llamaría “generación del software”.

1.3.1. Ingeniería de software

Se dice de la “disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad”.

Según la definición del IEEEⁱⁱⁱ "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según Lewis "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones eficaces en costo, a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos"^{iv}.

La ingeniería del software es el desarrollo, operación, mantenimiento del software de forma sistemática, disciplinada, cuantificable, y el estudio de los métodos que se aplican para generarlo.

El término ingeniería de software fue utilizado por primera vez por Fritz Bauer en la primera conferencia sobre desarrollo de software patrocinada por el Comité de Ciencia de la OTAN celebrada en Garmisch, Alemania, en octubre de 1968, y puede definirse según Alan Davis como “la aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios”

Por ende, la Ingeniería del Software se podría definir como *el establecimiento y aplicación de principios de la Ingeniería para obtener software*, teniendo en cuenta factores como el coste económico, la fiabilidad del sistema y el funcionamiento eficiente para satisfacer las necesidades del usuario.

Prestigiosos autores definen a la Ingeniería de Software como:

- ✓ Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkovitz, 1978)
- ✓ Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como Desarrollo de Software o Producción de Software (Bohem, 1976).
- ✓ Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).
- ✓ Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software (IEEE, 1993).

En el 2004, en los Estados Unidos, la Oficina de Estadísticas del Trabajo (U. S. Bureau of Labor Statistics) contó 760.840 ingenieros de software de computadora. El término "ingeniero de software", sin embargo, se utiliza en forma genérica en el ambiente empresarial, y no todos los ingenieros de software poseen realmente títulos de Ingeniería de universidades reconocidas.

Algunos autores consideran que *Desarrollo de Software* es un término más apropiado que *Ingeniería de Software* (IS) para el proceso de crear software. Personas como Pete McBreen^v cree que el término IS implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software.

1.3.1.1. Crisis del software

La crisis del software se fundamentó, debido a que en la creación del mismo no se obtenían los resultados deseados, este proceso demandaba grandes costos y poca flexibilidad.

Edsger Dijkstra^{vi} llamó "*crisis del software*" al período que va desde 1965 a 1985, debido a la demanda creciente de sistemas de computación complejos asociado a la falta de métodos y recursos formales necesarios para el desarrollo. La opinión que prevalecía hacia los años '70 era que uno debe primero escribir un programa y seguidamente proporcionar una prueba matemática de su corrección, Dijkstra objetó que las pruebas que resultan son largas e incómodas, y que la prueba no da ninguna comprensión de cómo se desarrolló el programa, propone la verificación formal, un método alternativo es la derivación de programas, desarrollar prueba y programa conjuntamente.

En esa época muchos grandes sistemas fracasaron por la falta de previsibilidad y estudios de costo-beneficio, por ende, se dejaba de lado un postulado fundamental de la ingeniería: ya que "*una actividad de ingeniería, por lo general, está limitada a un tiempo y recursos dados por proyectos*"

Esta crisis se superó en gran medida porque se estaban haciendo progresos en los procesos de diseño y metodologías tales como:

- ✓ Programación estructurada
- ✓ Programación Orientada a Objetos
- ✓ Programación Orientada a los Aspectos
- ✓ El surgimiento de las herramientas CASE
- ✓ El lenguaje de programación ADA
- ✓ La documentación
- ✓ Los estándares
- ✓ El lenguaje UML

La incorporación de métodos formales al desarrollo de software, hacía suponer que la industria sería predecible como lo son otras ramas de la ingeniería.

Todas estas metodologías se anunciaban como la solución a los problemas de la ingeniería del software, la llamada "bala de plata" (silver bullet) - concepto que nace del folklore en el cual se dice que el "hombre lobo" parece inocente pero es un monstruo que aterroriza, y que para matarlo mágicamente se necesita una "bala de plata"- para Fredrick P. Brooks, Jr.^{vii} *"el típico proyecto de software, por lo menos visto por el gerente no técnico, tiene algo de éste carácter: usualmente es inocente y simple. Pero puede transformarse en un monstruo de programas atrasados, presupuestos reventados y productos defectuosos. De modo que oímos gritos desesperados pidiendo balas de plata- algo que haga bajar los costos de software tan rápido como los costos de hardware. Pero no se ve en ningún lugar una bala de plata. No hay ningún desarrollo, ni en tecnología ni en técnicas de gestión, que por sí sólo prometa ni siquiera una mejora en un orden de magnitud en productividad, en fiabilidad, en simplicidad...No sólo no hay balas de plata a la vista, sino que la misma naturaleza del software impide que las haya",* luego, su propuesta es que cada organización de software debe determinar y proclamar que para su éxito los diseñadores brillantes son tan importantes como los gerentes brillantes, y se puede esperar que ellos sean cuidados y recompensados en forma similar.

1.3.1.1.1. El proceso de Ingeniería de software

Ivar Hjalmar Jacobson define el proceso de Ingeniería de Software como *"un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad"*. El proceso de desarrollo de software *"es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo"*. Concretamente *"define quién está haciendo, qué, cuándo hacerlo y cómo alcanzar un cierto objetivo"*.

Jacobson pensó que era necesario hacer planes de desarrollo de software, inventó los casos de uso como una forma de especificar los requisitos funcionales de software, creó el Lenguaje Unificado de Modelado (UML) y posteriormente desarrolló el Proceso Unificado Racional (RUP).

En noviembre de 2005, Jacobson anunció la Essential Unified Process (EssUP), una nueva "práctica" centrada en el proceso de desarrollo de software. Se trata de un nuevo comienzo a la integración de prácticas eficaces de entre los tres principales campos de proceso: el proceso unificado, los métodos ágiles y el proceso de madurez. Cada uno de ellos contribuye diferentes capacidades: estructura, agilidad y mejora de procesos.

El proceso de desarrollo de software requiere de un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el **ciclo de vida del software** que comprende las etapas por las que pasa un proyecto de software desde que es concebido, hasta que está listo para usarse. Típicamente, incluye las siguientes etapas: toma o elicitación de requisitos, análisis, diseño, desarrollo, pruebas (validación, aseguramiento de la calidad), instalación (implementación), uso, mantenimiento y actualización.

1.3.2. El surgimiento de la Web

La aparición de los servicios Web, trajo consigo un problema latente a resolver: la interoperabilidad. Aunque el intercambio de datos y procesamiento era posible en contextos tecnológicos homogéneos, se tornaba dificultoso en contextos heterogéneos, siendo los servicios web una solución tangible que surgía como una solución tecnológica fundamental, clave en este cambio de paradigma.

La conjunción de hipertexto y multimedia, por un lado, y la popularización de la Web, por otro, generaron desafíos que los métodos tradicionales no eran capaces, aparentemente, de resolver. Entre ellos destaca el diseño y control de la navegación por espacios complejos de

información, que ha llevado a los métodos de diseño de hipermedia y de la llamada Ingeniería Web a presentar el modelo navegacional como la gran diferencia frente a los métodos tradicionales. El modelo navegacional es su principal aportación. Este se deriva del modelo de datos o estructural e incluye como principal elemento las llamadas estructuras de acceso que sirven para indicar los caminos navegacionales disponibles en la aplicación. Aunque es discutible que la navegación sea exclusiva de la hipermedia o la Web (en cualquier aplicación existe navegación, tal vez no basada en enlaces, pero sí apoyada en otro tipo de elementos de interfaz de usuario), un mérito indiscutible de los métodos de Ingeniería Web ha sido explicitar la necesidad de un diseño cuidadoso de la misma.

El reconocimiento de la comunidad de Ingeniería de Software de que las mismas incluyen aspectos diferentes que las aplicadas a Ingeniería de Software, trajo aparejado el surgimiento de la Ingeniería Web, disciplina que propone extensiones y adecuaciones a los métodos y modelos tradicionales, para el modelado conceptual de este tipo de aplicaciones.

1.3.3. Ingeniería Web como un nuevo paradigma

Para Thomas Kuhn, un *paradigma* es un "conjunto de prácticas que definen una disciplina científica durante un período específico de tiempo". "Las sucesivas transiciones de un paradigma a otro vía alguna revolución, es el patrón de desarrollo usual de la ciencia madura"^{viii}.

Probablemente el uso más común de paradigma, implique el concepto de "cosmovisión" como el conjunto de experiencias, creencias y valores que afectan la forma en que un individuo percibe la realidad y la forma en que responde a esa percepción.

Para las ciencias sociales, el término se usa para describir el conjunto de experiencias, creencias y valores que afectan la forma en que un individuo percibe la realidad y la forma en que responde a esa percepción.

Los investigadores sociales han adoptado la frase de Kuhn: "cambio de paradigma", para remarcar un cambio en la forma en que una determinada sociedad organiza e interpreta la realidad.

Un "paradigma dominante" se refiere a los valores o sistemas de pensamiento en una sociedad estable, en un momento determinado.

Los paradigmas dominantes son compartidos por el trasfondo cultural de la comunidad y por el contexto histórico del momento.

Las condiciones que facilitan que un sistema de pensamiento pueda convertirse en un paradigma dominante son:

- ✓ Organizaciones profesionales que legitiman el paradigma.
- ✓ Líderes sociales que lo introducen y promueven.
- ✓ Periodismo que escribe acerca del sistema de pensamiento, legitimándolo al mismo tiempo que difunden el paradigma.
- ✓ Agencias gubernamentales que lo oficializan.
- ✓ Educadores que lo propagan al enseñar a sus alumnos.
- ✓ Conferencistas ávidos de discutir las ideas centrales del paradigma.

Adam Smith define al paradigma como "un conjunto compartido de suposiciones. Es la manera como percibimos el mundo: Agua para el pez. El paradigma nos explica el mundo y nos ayuda a predecir su comportamiento".

La nota que hace Smith sobre la predicción es de suma importancia porque allí está la clave en cuanto al valor de asumir los cambios de paradigma en su dimensión educativa, prospectiva y holística, es decir, en lo que tiene que ver con el logro de habilidades para asumir *el futuro y el cambio*.

Precisamente, la Ingeniería Web, cumple las condiciones establecidas por estos autores, ya que busca el logro de habilidades para asumir el futuro y el cambio que Internet impone además de cumplir con las condiciones de paradigma dominante.

1.4. Ingeniería Web

La Ingeniería Web (IWeb) aplica *“sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad”*.

Actualmente existen millones de sitios web que cubren diferentes necesidades de la vida, seguramente la comunicación es la más importante teniendo en cuenta que las distancias pasaron a un segundo plano.

El avance de Internet y las nuevas tecnologías de comunicación, marcan una nueva tendencia en el mercado del software. La exigencia de los usuarios, cada vez más numerosos, hizo que Internet creciera en forma acelerada y desprolija, produciendo un impacto social, económico y político impensado.

El nacimiento de las aplicaciones Web (WebApp) no se hizo esperar, aquellos sitios que en un comienzo fueron sólo páginas informativas, se vieron obligadas a brindar a sus visitantes algún tipo de servicio que conjugue páginas planas con datos almacenados. Todo este proceso, caótico, dio lugar al nacimiento de sitios no planificados en los que no se empleó una metodología apropiada.

Se venden millones de dólares por medio de Internet y de cada 100 sitios web, se estima que apenas un 10% presenta un buen desarrollo en función de su presentación y propósito, y apenas un 2% parecen maximizar el aprovechamiento del recurso. A pesar de ello, el crecimiento no es proporcional entre cantidad y calidad, los webmasters (personas responsables de la mantención y programación de un sitio Web) y creadores de páginas amateur crean sitios con enormes deficiencias en el área técnica, estética y práctica, y por la falta de investigación y educación de nivel, parece que esta será la media por mucho tiempo.

Por otro lado, se debe tener en cuenta que Internet crece un 20% anual aproximadamente y que la ingeniería web pone a su alcance cada día más y mejores herramientas para aumentar su efectividad.

1.5. Ingeniería Web Vs. Ingeniería de Software

En muchas ocasiones, en el desarrollo de WebApps, se aplicaron (o se aplican), herramientas de Ingeniería de Software, sin tener en cuenta que contienen características especiales como: usabilidad, navegabilidad, seguridad, mantenimiento, escalabilidad, entre otras.

Roger Pressman indica que el uso de soluciones de Ingeniería de Software, conduciría a una “web enmarañada” que entraña un cúmulo de aplicaciones basadas en web mal desarrolladas y que tienen muy altas probabilidades de fracaso. Por lo que sugiere que para lograr mayor éxito en el desarrollo y la aplicación de sistemas basados en Web complejos y a gran escala, existe una apremiante necesidad de enfoques disciplinados y nuevos métodos y herramientas con qué desarrollar, desplegar y evaluar los sistemas y aplicaciones basados en Web.

Si bien se pueden utilizar herramientas de análisis propias de la ingeniería de software para desarrollar WebApps, existen diferencias entre un producto propio de ingeniería de software y una WebApp.

1.6. La Ingeniería Web obtiene un espacio

Las aplicaciones Web, hacen posible el acceso impensado a recursos desde lugares remotos. Por ello, es preciso que los desarrolladores presten atención a los requerimientos, brindar calidad, evaluar rendimiento, considerar sistemas escalables sin perder de vista el mantenimiento y correcta documentación. El desarrollo de WebApps llegó para quedarse y se

prevé un largo ciclo de vida, por lo que se puede afirmar que aquellas aplicaciones que no se desarrollen de forma apropiada, tienen alta probabilidad de fallos y bajo rendimiento.

1.6.1 Primeros pasos en Ingeniería Web

La siguiente tabla muestra los primeros pasos de la Ingeniería Web según el sitio WebEngineering.org:

1995/1996	Los Primeros Estudios sobre los problemas relacionados con la Ingeniería Web aparecen en las Conferencias de World Wide Web en 1995/1996. En la quinta Conferencia de la World Wide Web 1996 en París, Francia, Bebo Blanca organizó un tutorial sobre documentos de la ingeniería web. Gellersen, Wicke, y Gaedke han hablado del ciclo de vida de la Ingeniería Web en la sexta Conferencia de la World Wide Web en Santa Clara, CA, EE.UU..
1998- Primer Workshop de Ingeniería Web	En la 7ª Conferencia de la World Wide Web en Brisbane, Australia en 1998 San Murugesan organizó el primer taller sobre Ingeniería Web
Registración de WebEngineering.org	En 1997, la Página de Inicio WebEngineering se publicó y hosteó la Oficina de Telecooperation sede de la Universidad de Karlsruhe para los estudiantes interesados en esta investigación. El sitio ha evolucionado debido a varias solicitudes de investigadores, estudiantes y otros en los meses siguientes. En 1998 estas actividades y la demanda de la todavía joven, pero creciente comunidad de conducir a la inscripción de WebEngineering.org
Progreso 1998 - 2001	De 1997 a 2001, varios eventos se organizaron con enfoque en Ingeniería Web. Estos eventos incluyen la serie de talleres de Conferencias World Wide Web y de la Conferencia Internacional sobre Ingeniería de Software (CISE), y la Conferencia Internacional sobre Ciencia de Sistemas de la Minitracks en Hawaii'n.
2001 - Taller Internacional de Software Orientada a Tecnología Web	Daniel Schwabe en 2001 organizó el primer Taller Internacional sobre el Tecnología de Software Orientado a Web. La organización local fue hecha por Oscar Pastor y su grupo. El objetivo de este "sólo por invitación", estaba listo: Todo el mundo tenía que resolver el mismo escenario-problema común - el desarrollo de una conferencia de revisión del sistema.
2001 – Rediseño del sitio WebEngineering.org	En 2001, la gestión de las TI y el grupo de investigación Ingeniería Web de la Universidad de Karlsruhe WebEngineering.org rediseña el sitio de la Comunidad. El nuevo sitio fue desarrollado utilizando el componente basado en la tecnología WCML (WebComposition Markup Language) y la evolución de servicios orientados a la Web basado en arquitectura de aplicaciones (Evolutionbus). El nuevo sistema impulsado por la WCML3-Evolutionbus Runtime Engine apoyó XML, XPath, XLinks, XSL / T, WebDAV y Automatización (Scripting).
2002 - La Undécima Conferencia Internacional de la World Wide Web (WWW2002)	Por primera vez la Ingeniería Web se convirtió en la parte principal del programa. El avance de la Ingeniería Web fue un éxito de documentos siguiendo las pistas de documentos co-presididos por Yogesh Deshpande, Martin Gaedke, San Murugesan y Daniel Schwabe. Los procedimientos WWW2002 están disponibles en línea. Estudios de Ingeniería de la trayectoria Web están marcadas con códigos de la sesión 1 a WE-WE-4.
2002 - Journal de Ingeniería Web (JWE)	En 2002, la primera revista internacional relacionada con la Ingeniería Web ha sido publicada por Prensa Rinton, NJ, EE.UU.. El Diario de Ingeniería Web (JWE) proporciona un foro para la distribución de información en todas las áreas de Ingeniería Web. Son publicados artículos originales, artículos de estudio, exámenes, tutorías, perspectivas, y correspondencia. Su contenido aborda importantes

	cuestiones y problemas, y las posibles soluciones, en el desarrollo de aplicaciones basadas en Web.
2003 - Duodécima Conferencia Internacional de World Wide Web (WWW2003)	La Ingeniería Web se convirtió en un espacio de referencia del avance de papers en la conferencia WWW2003 - demostrando la necesidad de una emergente disciplina en el desarrollo, funcionamiento y evolución de aplicaciones Web.
Relanzamiento de webengineering.org	En mayo de 2005, el IT-Management y grupo de investigación de ingeniería Web de la Universidad de Karlsruhe WebEngineering.org rediseñó el sitio de la Comunidad para ser completamente distribuido, basado en servicios Web y semánticamente habilitado. El nuevo sitio fue desarrollado sobre la base del Sistema libre de Vinculación de Servicio WebComposition (WSLS).
ICWE 2005	La Quinta Conferencia Internacional sobre Ingeniería Web (ICWE2005) reunió a la comunidad internacional de expertos en Ingeniería Web de investigación, práctica y educación. Además de interesantes charlas, presentaciones, discusiones y demostraciones, la comunidad inició el proceso de establecimiento de una sociedad internacional en Ingeniería Web.

Tabla 1 – Histórico de la Ingeniería Web

Capítulo 2 – Metodologías de la Iweb

2.1. Introducción

De hecho la WWW (World Wide Web) se ha establecido como una plataforma común para el desarrollo de sistemas de software. Las aplicaciones Web se han generalizado y se encuentran desarrollos importantes en diferentes ámbitos de la vida humana tales como la economía, industria, educación, administración pública, etc. Estas aplicaciones requieren de un análisis especial para el adecuado desarrollo.

En 1998, Roger Pressman moderó una mesa redonda virtual con representantes de la ingeniería de software tradicional y del desarrollo de software basado exclusivamente en Internet. El debate principalmente se centró en discutir si valía la pena aplicar un proceso de ingeniería a las aplicaciones con base en internet, o qué características tenían éstas que justificaran el no utilizarlo. La conclusión general fue que aplicar un proceso de ingeniería nunca es una mala idea pero que éste debería adaptarse a los requerimientos de cambio continuo y rapidez, siempre presentes, en el proceso de desarrollo Web. De iniciativas como ésta y de otras como la organización de congresos y talleres especializados en el desarrollo para la Web, surge el nacimiento de una nueva disciplina denominada Ingeniería Web.

El desarrollo de Aplicaciones Web (WebApp) es distinto respecto al desarrollo de software de otras categorías, estas diferencias son planteadas por Powell y referenciadas en Pressman en el año 2003: *“implica una mezcla de publicación impresa y desarrollo de software, marketing e informática, de comunicaciones internas y relaciones externas, y de arte y tecnología”*

Por tanto, el desarrollo de un WebApp es un esfuerzo multidisciplinario debido al manejo de múltiples formatos, con una mezcla de atención social para saber su efecto ético y legal.

En las primeras etapas de desarrollo de la Web, como ya se mencionó, parecía que las aplicaciones eran caóticas, carecían de estructura, lo que evidenciaba la falta de métodos de desarrollo adecuados, el desarrollo de aplicaciones Web requiere de un proceso estructurado y sistemático.

La ingeniería de software proporciona herramientas de análisis y diseño necesarias para que los desarrolladores puedan representar de forma abstracta un problema y a partir de ésta generar una solución informática, pero los métodos de análisis y diseño de WebApps contienen esos objetivos, más los añadidos por la dimensión de hipermedia como la navegación a través de la información y su presentación.

2.2. Proceso de la IWeb

La inmediatez, evolución y crecimiento continuos, son características de las aplicaciones Web, esto nos lleva a un proceso incremental y evolutivo, que permite que el usuario se involucre activamente, facilitando el desarrollo de productos que se ajustan a sus requerimientos.

Pressman enumera siete actividades que forman parte del proceso de la IWeb y que son aplicables a cualquier WebApp independientemente de su tamaño y complejidad. Las mismas son:

1. La **Formulación** identifica objetivos y establece el alcance de la primera entrega.
2. La **Planificación** genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.
3. El **Análisis** especifica los requerimientos e identifica el contenido.
4. La **Modelización** se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y

servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.

5. En la **Generación de Páginas** se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación: las páginas.
6. El **Test** busca errores a todos los niveles: contenido, funcional, navegacional, rendimiento, etc. El hecho de que las aplicaciones residan en la red, y que interoperen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil.
7. Finalmente, el resultado es sometido a la **Evaluación del Cliente**.

Una de las tareas colaterales que forman parte del proceso es el **Control y Garantía de la Calidad**. Este proceso, que se cumple en la ingeniería de software tradicional, implica actividades tales como: establecimiento y supervisión de estándares, revisiones técnicas formales, análisis, seguimiento y registro de informes, etc, son igualmente aplicables a la Ingeniería Web. Sin embargo, en la Web toman especial relevancia para valorar la calidad, aspectos como: **Usabilidad, Funcionalidad, Fiabilidad, Seguridad, Eficiencia y Mantenibilidad**^x.

2.3. Métodos de la Ingeniería Web

Los métodos de la Ingeniería Web definen las etapas y actividades necesarias para efectuar la construcción completa de una aplicación Web.

El principio subyacente en todos ellos es que *una aplicación Web debe desarrollarse partiendo de una descripción precisa en forma de un Esquema Conceptual que se transforma a una representación software, mediante un conjunto de correspondencias entre las abstracciones conceptuales que constituyen su esquema conceptual y los componentes software.*

En menor o mayor medida y a veces con diferentes nombres o sub-fases, la mayoría de los métodos coinciden en las siguientes etapas:

1. **Diseño Conceptual**: Trata de la especificación del dominio del problema, a través de la definición de datos y sus relaciones.
2. **Diseño Navegacional**: Establece los caminos de acceso a la información y sus permisos de visibilidad.
3. **Diseño de la presentación o diseño de Interfaz**: Define cómo se muestra la información en la interfaz de usuario.
4. **Implementación**: Es la construcción del software a partir de los artefactos generados en las etapas previas.

El presente capítulo muestra algunos métodos de Ingeniería Web, a los que se aplicará el mismo ejemplo para representarlos y compararlos.

2.3.1. Ejemplo con el que se trabajará en este capítulo:

Se trata de una institución que dicta cursos con las siguientes características:

1. Modalidad: a distancia y presencial.
2. Cada curso se dicta con profesores de importante trayectoria.
3. Los alumnos deberán poder consultar:
 - a. Contenidos del curso
 - b. Fecha de dictado
 - c. Si es en modalidad presencial: sala en que se dicta
 - d. Si es modalidad a distancia, detalles de la página, clave de acceso
 - e. Docente que dicta el curso, Curriculum Vitae.
4. Debe además poder modificar su perfil.

2.3.2. Hypertext Design Model (HDM)

Modelo de Diseño de Hipertexto creado en 1991 por Franca Garzotto, Paolo Paolini y Daniel Schwabe para la creación de aplicaciones hipermediales.

En este método se distinguen dos etapas en el diseño de aplicaciones web:

1. El diseño a gran escala (*design in the large*), se refiere al diseño global y a los aspectos estructurales de la aplicación, trata la definición de las relaciones conceptuales entre los nodos de la aplicación.
2. El diseño a pequeña escala (*design in the small*), se refiere al desarrollo del contenido de los nodos de hipermedia, y está relacionado con la implementación de éstos. Trata de solucionar problemas como la obtención de la información desde una base de datos, y con qué herramientas de desarrollo se programará, etc.

2.3.2.1. Proceso

El proceso de desarrollo en HDM está integrado por los siguientes pasos:

1. Identificar el grupo de entidades del mundo real y sus componentes. Las relaciones entre una entidad y sus componentes determinan la navegación estructural.
2. Identificar dentro de los objetos del mundo aquellos que tienen estructura y conexiones similares, los que son del mismo tipo.
3. Determinar el conjunto de colecciones de objetos y sus miembros. En este paso se diseñan los enlaces de aplicación derivados de las colecciones.
4. Se determina si el tipo de navegación de cada colección es mediante un índice o una visita guiada.

2.3.2.2. Componentes

a) Nodos

La información está agrupada en nodos. Un nodo es una unidad de información que contiene texto o elementos multimedia como video, sonido o imágenes. Los nodos se representan mediante óvalos y son identificados por un nombre.

Tipos de nodos en HDM:

1. **Unidad:** es un nodo que no se compone por ningún otro y constituye una hoja en la jerarquía.
2. **Componente:** integrado por un conjunto de unidades o componentes, tienen una relación de pertenencia con el componente superior. Los componentes forman jerarquías.
3. **Entidad:** es una estructura de información que describe un objeto abstracto o real parte del dominio de la aplicación. Una entidad define un tipo, y las entidades del mismo tipo comparten la misma estructura y conexiones con las demás, es un componente raíz.

b) Enlaces

Los nodos se relacionan mediante enlaces. Un enlace conecta dos nodos de forma unidireccional y está asociado a un elemento específico dentro del nodo donde se muestra.

Los enlaces pueden ser de tres tipos:

1. Los **estructurales:** que sirven para conectar componentes de una misma entidad. Conectan entidades, componentes y unidades de una jerarquía. Este tipo de enlaces se identifican por una flecha triangular.
2. De **aplicación:** conectan nodos de distintas jerarquías, se representan por una flecha normal.
3. De **perspectiva:** corresponde a una vista de la aplicación. Cada perspectiva define un conjunto de propiedades que se visualizan en la jerarquía de componentes. Se definen a nivel de entidad, cada componente debe tener la definición de propiedades.

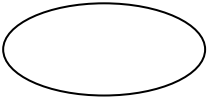


			Nombre={valor1, valor2,...}
Nodos	Enlace Estructural	Enlace de Aplicación	Enlace de Perspectiva

Figura 1 – Notación HDM

c) Elementos de Navegación

Los elementos de navegación son **colecciones y estructuras de acceso**.

1. *Colección*: es un conjunto de enlaces que apunta a nodos hipermedia. Las colecciones pueden ser de dos tipos:
 - a. Un *índice* permite navegar directamente a los nodos que pertenecen a la colección.
 - b. Una *visita guiada*: muestra la colección de objetos a través de un recorrido secuencial hacia adelante y atrás.
2. Las *estructuras de acceso* son componentes y no tienen un símbolo para especificarlas, se indican en el nombre del nodo como etiqueta.

2.3.2.3. Aplicación del Ejemplo

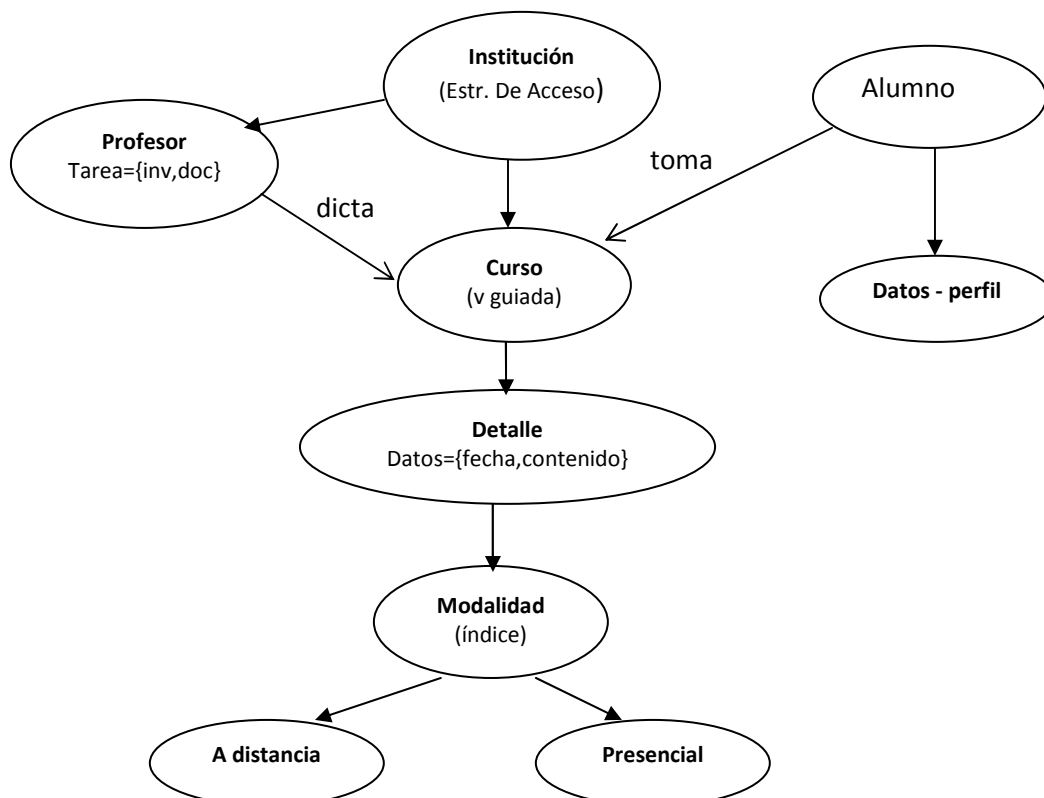


Figura 2 – Ejemplo en HDM

Nodos:

- ✓ **Entidades:** Institución, alumno
- ✓ **Componentes:** Institución compuesta por cursos y profesores. Curso compuesta por detalles y modalidad que a su vez pueden ser a distancia o presenciales.
- ✓ **Unidad:** Datos del alumno, que puede ser su perfil.

Enlaces:

- ✓ **Estructurales:** La entidad Institución posee un enlace estructural con los componentes curso, detalles y modalidad cuyas unidades son presencial o a distancia.
- ✓ **De Aplicación:** Profesor que dicta curso, Alumno que toma el curso
- ✓ **De perspectiva:** Profesor que puede ser investigador y docente, Detalle del curso que posee fecha de inicio y contenido.

Elementos de Navegación:

- ✓ **Colección:** Curso con una visita guiada, modalidad con un índice.
- ✓ **Estructura de Acceso:** Institución

2.3.2.4. Consideraciones respecto del método HDM

Carece de una notación clara para representar las estructuras de acceso ya que todos son óvalos. Su modelo de datos no es estándar, no es relacional, ni orientado a objetos.

HDM se centra en el diseño a gran escala. El diseño a pequeña escala prácticamente no es abordado en el método y se limita a la asignación de contenido a los nodos.

HDM propuso las estructuras de acceso, que luego fueron usados por métodos posteriores.

Distinguió claramente la composición de los nodos hipermedia, y la necesidad de diferentes vistas para una misma entidad.

2.3.3. Relationship Management Methodology (RMM)

Este método fue creado por ISAKOWITZ, T.; STOHR, E.A.; BALASUBRAMANIAN^{xi}, según sus autores, RMM está orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos. La hipermedia es un medio para administrar relaciones entre objetos de información, es una técnica utilizada principalmente para el diseño y construcción de aplicaciones hipermedia.

Este modelo, posee como principales elementos: el modelo Entidad-Relación y el modelo RMDM (Relationship Management Data Model) basado en el modelo HDM.

Proporciona un medio para automatizar el desarrollo inicial y el proceso de actualización posterior. Se centra en el diseño, desarrollo y construcción de las fases del ciclo de vida de desarrollo de software.

Es una metodología para el desarrollo de aplicaciones Web que tienen una estructura regular definida mediante entidades y relaciones entre éstas.

2.3.3.1. Proceso

Etapas en el desarrollo de RMM:

- 1. Diseño Entidad-Relación (E-R):** se representa el dominio de la aplicación a través de un diagrama E-R [Che85] siendo las primitivas del dominio (domain primitives): *entidades*, *atributos* y *relaciones*.
Las *entidades* y sus *atributos* representan objetos abstractos o físicos y sus propiedades.
Las relaciones describen las asociaciones entre los diferentes tipos de entidades. Las asociaciones pueden ser de *uno a uno* (1-1) o de *uno a muchos* (1-N). Las relaciones de *muchos a muchos* (N-M) se factorizan en dos relaciones de uno a muchos, para facilitar su conversión a estructuras de acceso en la etapa 3.
- 2. Diseño de Secciones (Slice Design):** Este paso consiste en dividir una entidad en secciones significativas (slices) y organizarlas dentro de una red de hipertexto.

3. **Diseño Navegacional:** En esta etapa se diseñan los caminos que existirán en el grafo del documento hipermedia. Se analizan las relaciones y de acuerdo a ellas, se establece el tipo de estructura de acceso:
 - a. Relaciones 1-1: hiperenlaces.
 - b. Relación 1-N: índices o visitas guiadas. Se aconseja utilizar una visita guiada cuando el número de instancias es menor a diez, y cuando no hay una clave única para diseñar un índice. Cuando el conjunto de instancias es grande y existe una clave discriminadora se recomienda utilizar un índice en lugar de una visita guiada.
 - c. Menú: grupos de entidades de acuerdo al interés o comodidad del usuario final, sirve para construir los caminos de acceso desde un nodo origen.
4. **Diseño del protocolo de conversión:** Se establecen un conjunto de reglas de conversión de los elementos de RMM hacia la plataforma tecnológica de implementación. Esta etapa es realizada manualmente por los programadores.
5. **Diseño de la interfaz de usuario:** Consiste en el diseño de la distribución de los elementos en las interfaces, es decir, dónde se localizarán las etiquetas, atributos y enlaces, y qué apariencia deben tener cada uno de ellos.
6. **Comportamiento en tiempo de ejecución:** Trata sobre el comportamiento de los nodos, y si presentarán un historial de la navegación, un enlace hacia la página anterior, si se deben memorizar las páginas visitadas, o si los nodos se deben computar cada vez, etc.
7. **Construcción y pruebas:** Se siguen las técnicas establecidas en la ingeniería de software tradicional. En el caso de aplicaciones hipermedia se debe tener especial cuidado en las pruebas de los caminos navegacionales.

En cuanto al diseño Navegacional

La navegación es modelada mediante enlaces unidireccionales, enlaces bidireccionales, y estructuras de acceso que pueden ser índices, visitas guiadas, índice-visita-guiada y grupos

1. **Enlaces:** Existen dos tipos de enlaces:
 - a. *enlaces estructurales:* se denominan así a aquellos que conectan una entidad con sus secciones, o secciones de una misma entidad, las conexiones pueden ser bidireccionales o unidireccionales y
 - b. *enlaces:* que son unidireccionales y conectan entidades o estructuras de acceso entre sí.
2. **Grupo:** Es una construcción similar a un menú; está formado por una lista de enlaces.
3. **Índice:** es un grupo que contiene una lista de enlaces hacia instancias de entidades a las que provee un acceso directo.
4. **Visita guiada:** es un camino lineal a través de una colección de instancias de entidades, permite el movimiento a la instancia anterior y a la siguiente. Variantes de visitas guiadas: (1)*visita guiada circular:* en donde el último elemento tiene como siguiente al primero; (2)*visita guiada con retorno* al nodo principal, en la que hay una descripción de la visita guiada y es el nodo inicial y final de la misma; (3)*índice-visita guiada:* es una combinación de índice y visita guiada, posee diversos nodos de entrada. Además, es posible cualificar índices y visitas guiadas con predicados lógicos para determinar el conjunto de instancias que son accesibles desde ellos.

2.3.3.2. Notación

En la siguiente figura se muestra la notación utilizada en el método RMM.

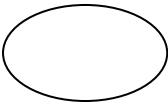
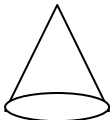
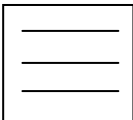
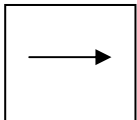
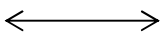
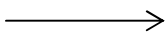
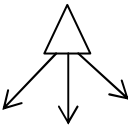
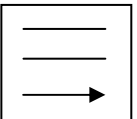
			
Entidad	Sección	Indice	Visita guiada
			
Enlace Bidireccional	Enlace Unidireccional	Grupo	Indice-Visita Guiada

Figura 3 – Notación del método RMM

2.3.3.3. Modelo de Datos de Administración de Relaciones (Relationship Management Data Model, RMDM).

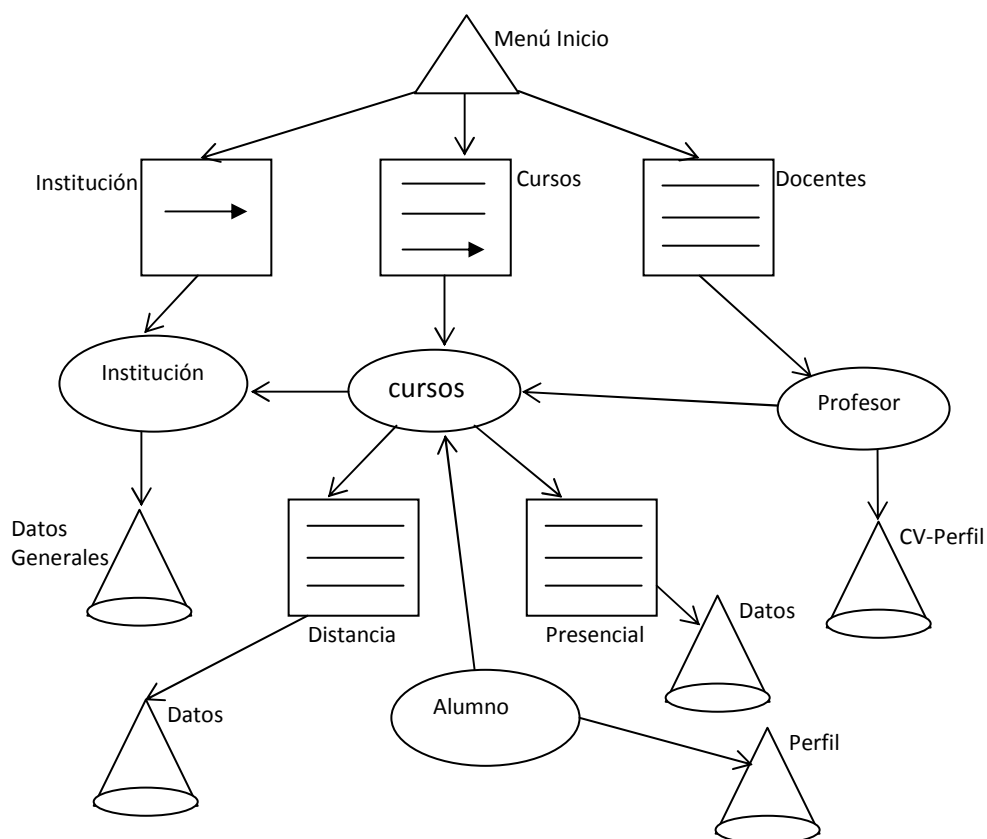


Figura 4 - RMDM

Estos modelos se construyen con primitivas de RMM, algunos ejemplos a los que se puede aplicar este modelo son catálogos y bases de datos.

El modelo RMDM de la figura 4 se explica de la siguiente manera, a partir del menú principal:

- ✓ Visita guiada para institución y un enlace estructural donde se presentan datos generales de la institución (oferta académica, características, fotos, etc)
- ✓ Índice-visita-guiada por los cursos, índice de acuerdo a la modalidad con un enlace estructural en donde se muestran datos generales del curso (fecha de inicio, contenidos, docente, etc)
- ✓ Índice de profesores que dictan el curso con un enlace estructural hacia su CV, perfil, datos generales.

- ✓ Enlace simple hacia alumno, y un enlace estructural hacia su perfil.

2.3.3.4. Consideraciones respecto del modelo RMM

RMM fue una mejora respecto a HDM, porque presenta una notación más completa y utiliza el modelo Entidad-Relación para especificar el dominio del problema. Incorpora los conceptos de secciones y de relación estructural en una metáfora hipermedia.

En RMM, el modelo hipermedia retoma los elementos: *enlace*, *índice* y *visitas guiadas* de HDM enriqueciéndolos con capacidades condicionales.

2.3.4. Object Oriented Hypermedia DesignMethod (OOHDM)

OOHDM (Modelo de Diseño de Hipermedia Orientado a Objetos - Object Oriented Hypermedia Design Model), fue creado en 1996 por Daniel Schwabe (Pontificia Universidade Católica do Rio de Janeiro) y Gustavo Rossi (Universidad Nacional de La Plata). Este método surge como una extensión del método HDM, a diferencia de HDM, se introduce el modelado orientado a objetos en el desarrollo de hipermedia.

En OOHDM se modela la navegación a través del diagrama de clases navegacionales y del diagrama de contextos.

Las etapas de Diseño Conceptual, Diseño Navegacional, Diseño de Interfases e Implementación son desarrolladas en un proceso que combina un diseño incremental, iterativo y basado en prototipos. El foco está puesto en enriquecer la interacción anterior.

Posee actividades separadas que permiten obtener diseños modulares y reusables. También se puede obtener un framework que contenga diseños específicos que se pueden usar en cada una de las etapas. Los desarrolladores podrán ir generando su propia biblioteca de recursos que podrán utilizar en futuros desarrollos, generando un léxico común que mejora la comunicación dentro del equipo.

Finalmente se buscan esquemas que permitan mapear las primitivas de diseño (nodos, links, contextos, etc.) en ambientes de implementación. En el caso de implementación en ambientes o lenguajes no orientados a objetos, no es directa, pero al ser un método orientado a objetos, el mapeo con tecnologías orientadas a objetos se ve facilitada.

2.3.4.1. Proceso

OOHDM considera que el desarrollo de una aplicación hipermedial se da en un proceso que posee cuatro actividades principales:

1. **Diseño conceptual:** Se construye un modelo del dominio de la aplicación, a través de las técnicas del modelado orientado a objetos, se puede partir de un modelo E/R.

Se identifican las clases y sus relaciones, que pueden ser de asociación, agregación, composición, y generalización-especialización. El resultado de esta etapa es un modelo estructural compuesto por clases, asociaciones y atributos, y es similar al diagrama de clases del Unified Modelling Language (UML).

2. **Diseño navegacional:** En OOHDM, la navegación es considerada un paso crítico en el diseño de aplicaciones de hipermedia, es construido como una vista del modelo conceptual.

En OOHDM, una aplicación es concebida como una vista navegacional del modelo conceptual. A partir de consultas en los modelos E/R (comportamiento que surge en el modelo conceptual) se pueden expresar diferentes vistas para diferentes perfiles de usuarios. Este procedimiento permite construir diferentes modelos navegacionales como una vista “caprichosa” del modelo conceptual y constituyen los Objetos Navegacionales.

Para construir la estructura navegacional se debe tener en cuenta:

- ✓ Nodos que serán navegables, establecer los atributos que poseen y sus relaciones (que son las que constituyen los Links).
- ✓ Contextos en que el usuario navegará para organizar el espacio navegacional.
- ✓ Vistas de los objetos navegacionales: de acuerdo al contexto en que se esté navegando.

- ✓ Estructuras que permitirán acceder a los nodos: ya sea iniciar recorridos dentro de alguno de los contextos definidos, realizar operaciones de búsqueda o armar páginas iniciales de la aplicación navegable.

Una de las mayores innovaciones de OOHDM es que el usuario no navega directamente dentro de las entidades conceptuales, sino en los Nodos (objetos navegacionales) que son construidos desde uno o varias entidades conceptuales. Entonces los atributos de los nodos se obtienen a partir de los atributos del modelo conceptual.

El modelo navegacional está integrado por: (a) el diagrama de *clases navegacionales* y (b) el diagrama de *contextos navegacionales*.

3. **Diseño de interfaz abstracta:** En esta etapa se define la forma en la que serán percibidos los objetos a través de la interfaz de usuario y también la apariencia que tendrán. La separación de diseño navegacional y de la interfaz de usuario permite dividir las tareas del desarrollo, así como tener diferentes interfaces para un mismo modelo navegacional. En OOHDM se utilizan vistas abstractas de datos (abstract data views, ADV). Mediante un ADV se representa la estructura estática de la interfaz, la composición de objetos y los eventos a los que responden.

4. **Implementación:** Es la última etapa, en la que, a partir de los modelos diseñados, se deben escoger las correspondencias con los objetos concretos de la plataforma de implementación. Es por lo tanto, una etapa totalmente dependiente de la plataforma de implementación escogida.

La siguiente figura muestra las etapas del proceso donde se presenta la posibilidad de obtener dos diseños navegacionales a partir de un modelo conceptual.

OOHDM, hace hincapié en la reutilización de componentes.

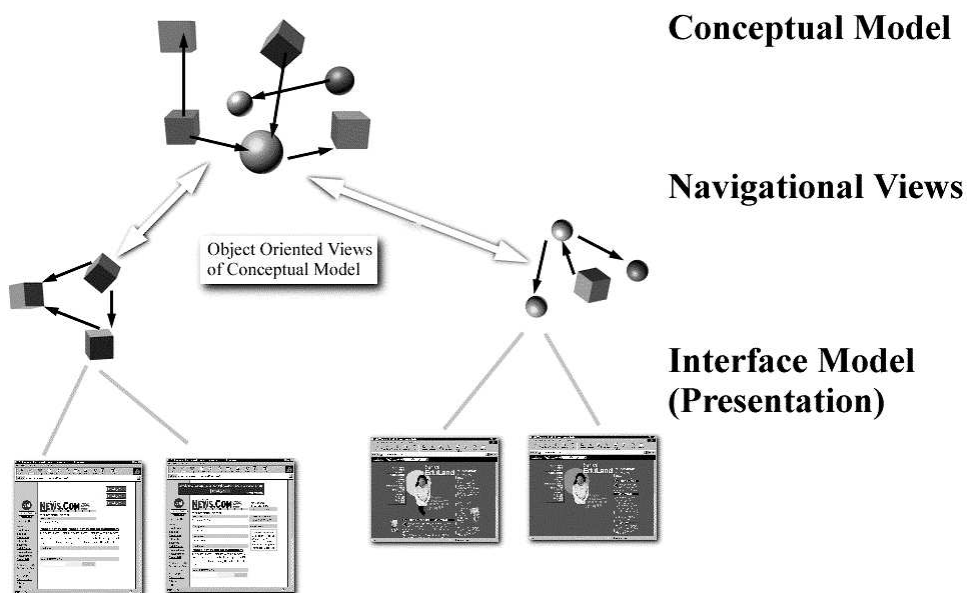


Figura 5 – OOHDM (extraído de “Developing Hypermedia Applications using OOHDM” Daniel Schwabe and Gustavo Rossi)

2.3.4.2. Aplicación del método OOHDM al ejemplo

Modelo Conceptual

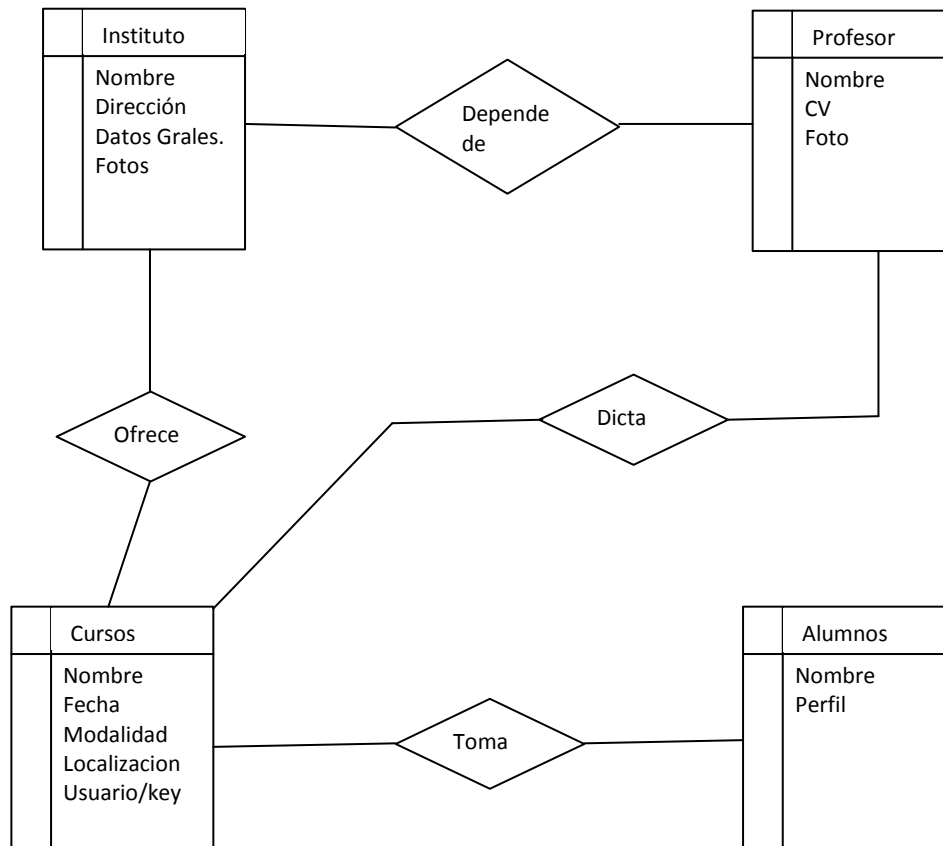


Figura 6 – Modelo Conceptual

Modelo Navegacional

a) Diagrama de clases navegacionales (figura 7)

En este caso se definen “caprichosamente” tres objetos navegacionales: Instituto, Curso, Alumno, teniendo en cuenta que el instituto desea promocionar cursos de acuerdo a la modalidad, docentes (por su trayectoria) y su establecimiento.

b) Diagrama de Contextos Navegacionales y estructuras de acceso (figura 8)

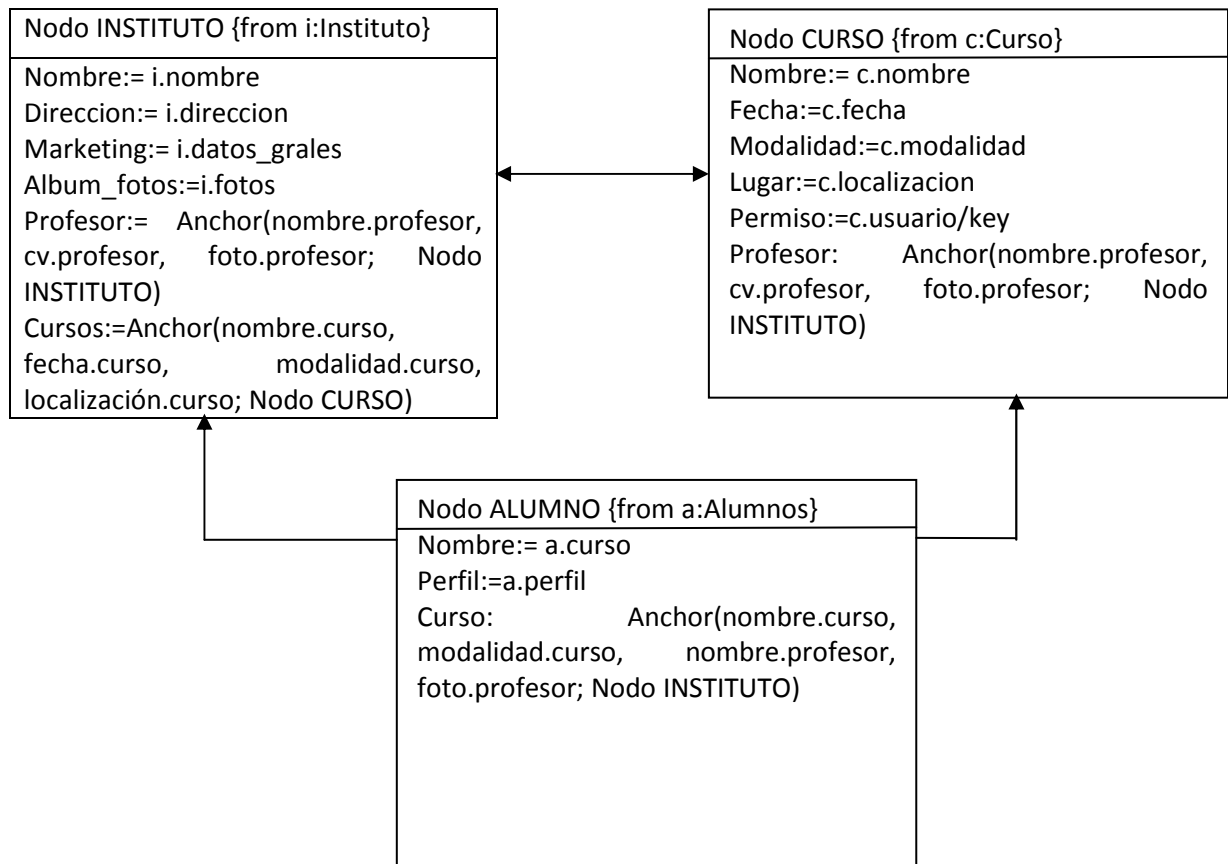


Figura 7 – Diagrama de Clases Navegacionales

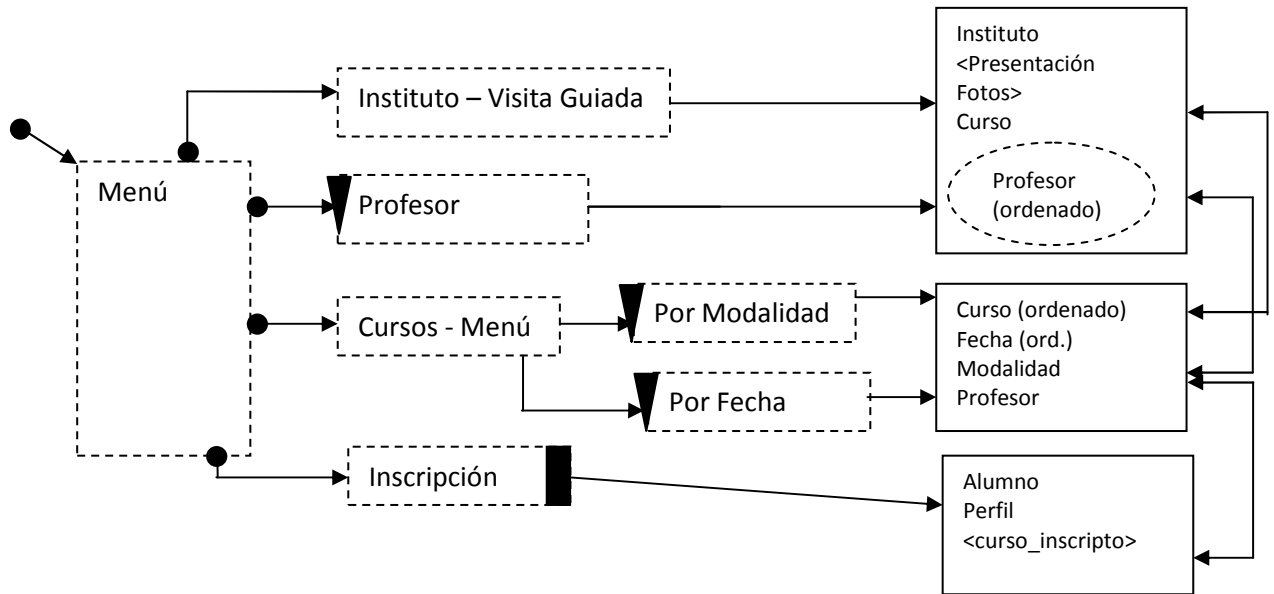


Figura 8 – Diagrama de Contextos Navegacionales

2.3.4.3. Consideraciones respecto del modelo OOHDM

En un principio, el método fue propuesto como solución a la carencia de abstracciones que tienen los métodos tradicionales de Ingeniería de Software para la especificación de aplicaciones que engloban la metáfora hipermedial al consultar y actualizar colecciones de datos en las bases de datos.

Schmid y Rossi extendieron, en el método, los Modelos conceptual y navegacional para incorporar actividades y entidades, así como nodos contenedores y primitivas de control de procesos para la especificación.

OOHDM es un método para modelar aplicaciones web de gran envergadura, que puede ser usado para modelar distintos tipos de aplicaciones navegables, sitios web, sistemas de información, presentaciones multimedia, etc.

2.3.5. Método OO-H

Este método fue desarrollado en el año 2000 por Jaime Gómez y Cristina Cachero de la Universidad de Alicante junto a Oscar Pastor de la Universidad Politécnica de Valencia. OO-H “es un modelo genérico que ofrece al diseñador la semántica y notación necesaria para el desarrollo de aplicaciones Web así como su conexión con la lógica de la aplicación existente”^{xii}. OO-H reconoce que además de la navegación que procede de la estructura de datos, hay otra derivada de la funcionalidad del sistema, y por ello, clasifica la navegación en estructural y semántica.

2.3.5.1. Proceso

1. **Análisis:** Los requisitos del usuario son plasmados en diagramas de casos de uso. A partir de estos diagramas, y a través de técnicas conocidas de análisis orientado a objetos, se deriva el modelo estructural, que se representa como un diagrama de clases UML.

En esta etapa, se establecen dos tipos de casos de uso dependiendo del flujo de la información: (1) *flujo simple*: en el que el caso de uso comprende solamente actividad de navegación; se estereotipa con la palabra clave <<navigation>> y (2) *flujo complejo*:

corresponde a un proceso que se refinará mediante diagramas adicionales pero que no se estereotipa.

Una vez establecidos los requisitos, el siguiente paso es el análisis del dominio del problema que se lleva a cabo mediante la construcción de un *modelo conceptual* que refleja la estructura del dominio.

2. **Definición de las unidades semánticas:** Se agrupan los casos de uso de acuerdo a los requisitos a las que se les denomina unidad semántica de navegación. Se establece en qué unidades semánticas participan cada grupo de usuarios y se las agrupa mediante un menú.
3. **Diseño navegacional:** Se define un conjunto de *Diagramas de Acceso Navegacional* (NAD) para cada tipo de usuario, incluyendo la información, servicios y rutas de navegación requeridos para la satisfacción de sus requisitos. El modelo navegacional, está formado por un conjunto de uno o más Diagramas de Acceso Navegacional (NAD's).

Este método tiene en cuenta (1) la navegación semántica que se deriva de los casos de uso en los que participa el usuario y (2) la navegación estructural que se deriva de las relaciones de los datos.

La navegación estructural no se refleja en un cambio en la intención de efectuar un cierto caso de uso. El orden de navegación entre las unidades semánticas se define mediante los enlaces semánticos. Este modelo es llamado modelo de análisis de la navegación.

Después se diseñan las clases navegacionales, que son una vista de las clases del modelo estructural. Un grupo de clases navegacionales usadas para efectuar un requisito es llamado destino navegacional. La relación entre las clases navegacionales dentro de un destino navegacional es a través de enlaces internos (I-link). En cambio, las relaciones entre clases de diferentes destinos navegacionales se relacionan a través de enlaces de travesía (T-link). El nodo inicial de un destino navegacional es indicado mediante un enlace de requisito (R-link). Los servicios que proporcionan las clases navegacionales son efectuados a través de los enlaces de servicio (S-link).

4. **Diseño de la presentación:** Se establece un Diagrama Abstracto de Presentación (APD). La presentación es diseñada con vistas abstractas de datos, las cuales definen una jerarquía de elementos gráficos que permiten diseñar la presentación de las clases navegacionales.

2.3.5.1. Consideraciones respecto del modelo OO-H

OO-H puede verse como una extensión a OOHDM con casos de uso y enlaces de servicio. OO-H, a través de casos de uso, y su técnica de encadenar la ejecución de éstos, llega a establecer un modelo parecido al de proceso.

2.3.6. Object Oriented Web Solution (OOWS)

OOWS [FVR+03] es una extensión al método OO-Method [Pas92], que permite capturar los requisitos funcionales de un sistema orientado a objetos para generar una especificación formal en el lenguaje OASIS [LRSP98]. A partir de la especificación formal, y mediante un compilador de modelos, se genera un prototipo del sistema. OOWS le proporciona a OO-M los modelos de navegación y presentación.

2.3.6.1. Proceso

El proceso de desarrollo de OOWS se divide en dos grandes fases: (1) modelado conceptual y (2) desarrollo de la solución. La siguiente figura muestra el proceso de desarrollo de éste método:

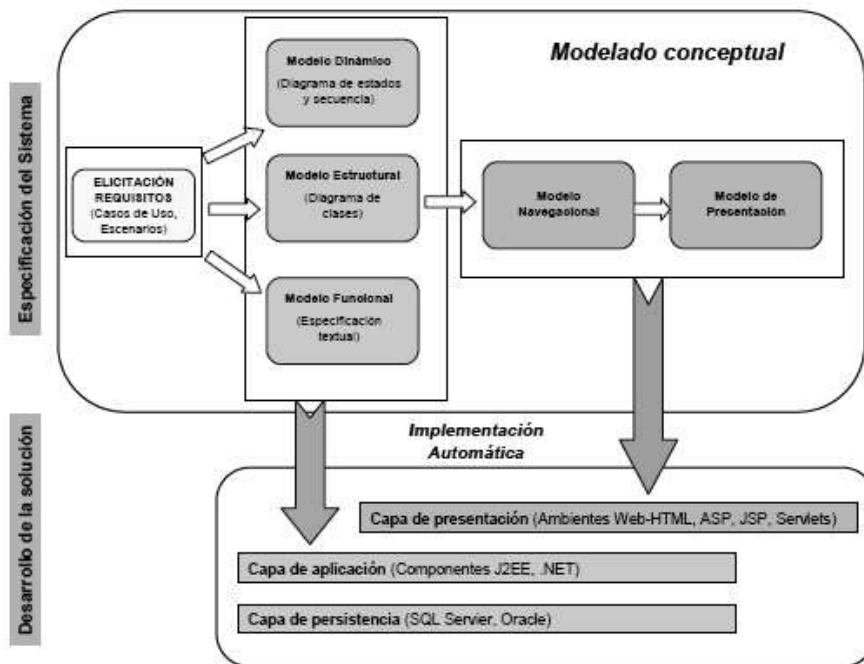


Figura 9 – Método OOWS (Extraído de “Desarrollo Dirigido por Modelos de Aplicaciones Web que integran Datos y Funcionalidad a partir de Servicios Web” de Ricardo Rafael Quintero Meza)

Etapas del Modelado Conceptual:

1. **Captura de requisitos funcionales:** que se realiza mediante diagramas de casos de uso y escenarios, que luego son usados para construir el modelo conceptual.
2. **Modelado conceptual clásico:** captura la estructura y el comportamiento del sistema desde los puntos de vista: (1) *estructural*, (2) *funcional* y (3) *dinámico*.
 - (1) *Estructural:* Se establece un *Diagrama de Clases* mediante la definición de la estructura del sistema que se logra estableciendo por un lado: sus clases, operaciones y atributos, y por otro: estableciendo las relaciones entre clases mediante especialización, asociación y agregación.
 - (2) *Funcional:* captura la semántica de los cambios de estado de los objetos para definir el efecto de los servicios, usando una especificación textual formal.
 - (3) *Dinámico:* se realizan dos tipos de diagramas (a) el *Diagramas de Transición de Estados*, una descripción de los diferentes ciclos de vida de cada clase del sistema y (b) el *Diagramas de Secuencia* constituido por las interacciones y comunicación entre objetos.
3. **Modelado de la navegación y de la presentación:** se establecen los requisitos de navegación por medio de un (1) *Diagrama de Usuarios* y (2) un *Modelo Navegacional*; y los requisitos de presentación por medio de (3) un *Modelo de Presentación*.
 - (1) *Diagrama de Usuarios:* se especifican los tipos de usuarios como *anónimos*, *genéricos*, *registrados*, y de acuerdo a ello se establecen los permisos de acceso al sistema.
 - (2) *Modelo Navegacional :* Este modelo se forma por una colección de Mapas Navegacionales. Un mapa es un grafo dirigido en el que los nodos son llamados (a) *Contextos Navegacionales* y las aristas (b) *Enlaces Navegacionales*. Estos mapas navegacionales, son representativos de la vista hipermedial del usuario sobre las clases, ofrece semántica de navegación y presentación.

El Modelo Navegacional captura en forma clara, los requisitos de navegación de la aplicación Web.

(a) Un *Contexto navegacional*: es una unidad de interacción que representa una vista de datos y su funcionalidad (atributos y operaciones de las clases), es representado por un paquete UML, posee el estereotipo <<Context>>.

Según la forma de ser accedidos, pueden ser: *Contextos de Exploración* o *Contexto de Secuencia*. El *Contexto de Exploración* se establece si son accesibles desde cualquier nodo y se indica con la letra “E” uno de los cuales se identifica como el *Contexto de inicio o Home* y es indicado con la letra “H”. El segundo caso: *Contexto de Secuencia* es indicado con la letra “S” y es cuando el contexto se alcanza a través de una secuencia de pasos de navegación (*Enlaces Navegacionales*).

(b) Los *Enlaces Navegacionales* representan pasos de navegación entre los contextos y se representan mediante asociaciones entre paquetes UML.

(3) *Modelo de Presentación*: Se establecen los requisitos de presentación de los Contextos Navegacionales, los cuales se especifican por medio de un conjunto de patrones que se asocian a las primitivas del contexto:

a. *De paginación de la información*: que permite definir la capacidad de desplazamiento vertical (scrolling) sobre la información. Todas las instancias se dividen en bloques lógicos, de tal forma que cada bloque es visible a la vez. Además se ofrecen mecanismos para avanzar y retroceder. El patrón se aplica a la clase directora, a la relación navegacional, a un índice o a un filtro. La información requerida por el patrón incluye: (1) la *Cardinalidad*, el número de instancias del bloque; (2) el *Modo de acceso*: secuencial, para acceder al bloque siguiente, previo, primero y último y aleatorio, para acceso directo o circular, para un comportamiento circular del conjunto de bloques.

b. *De ordenamiento*: que define un criterio de ordenamiento de la población (con valores ASC=ascendente o DESC=descendente) acorde al valor de uno o más atributos. Se puede aplicar a clases navegacionales, a estructuras de acceso o mecanismos de búsqueda.

c. *De distribución de la información*: con cuatro patrones básicos: registro, tabular, maestro-detalle y árbol. Se utiliza para indicar la forma en que se mostrará la información. Se puede aplicar a la clase directora (constituye la información principal que se recupera para el Contexto Navegacional) o a las relaciones navegacionales.

2.3.6.2. Consideraciones respecto del modelo OOWS

El método OOWS fue diseñado originalmente para la especificación y generación de aplicaciones Web centradas-en-datos. Para extender sus capacidades, de tal manera que pueda además integrar datos y funcionalidad de aplicaciones externas, se requiere que las aplicaciones Web que especifica y genera, cuenten con la facultad de producir y consumir funcionalidad mediante servicios Web.

El modelo de presentación en OOWS trata con algunas cuestiones como la paginación y ordenamiento de la información que pueden ser tratadas en las estructuras de acceso.

2.3.7. UML based Web Engineering (UWE)

UWE, desarrollado por Nora Koch, del Instituto de Informática de la Universität Manchen de Alemania, es un método de desarrollo de aplicaciones Web basado en UML estándar.

Además de la notación UML, utilizan el proceso unificado de desarrollo de software (Rational Unified Process o RUP) [BRJ99] como metodología para realizar aplicaciones hipermedia, por lo cual el proceso es iterativo e incremental. El método es muy similar a OOHD, y la principal diferencia radica en la notación.

La estrategia de diseño UWE se basa en modelos que se construyen durante la fase de análisis, principalmente el modelo conceptual y el modelo de procesos.

UWE introduce clases específicas de procesos como parte de un modelo separado, que ofrece una interfaz al modelo de navegación.

2.3.7.1. Proceso

1. *Análisis de requisitos*: El análisis de requisitos se expresa a través de la especificación de los casos de uso del sistema.
2. *Diseño conceptual*: En esta etapa se representa el dominio del problema con un diagrama de clases de UML. Los casos de uso sirven como entrada para elaborar tarjetas Clase-Responsabilidad-Colaborador (CRC), o para la identificación de verbos y sustantivos, entre otras técnicas, que permiten determinar las *clases, métodos y atributos*.
3. *Diseño navegacional*: tiene dos etapas (1) la *definición del espacio de navegación* y (2) el *diseño de las estructuras de navegación*.

- (1) La *definición del espacio de navegación* se trata de una vista del diagrama conceptual, se define mediante el diagrama de clases UML y se distinguen con el estereotipo NC.

Los Espacios navegacionales tienen las siguientes características:

- ✓ Las clases conceptuales que son importantes para el usuario, permanecen en el modelo navegacional.
 - ✓ Las clases que no se visitan, pero que contienen atributos importantes, no aparecen en el modelo navegacional, y sus atributos se muestran como parte de otras clases.
 - ✓ En el caso de vistas complejas se emplea Object Query Language para construirlas.
 - ✓ Para evitar caminos navegacionales profundos, se incorporan al modelo de navegación, asociaciones adicionales que están etiquetadas con un estereotipo y representan la navegación directa entre clases.
 - ✓ Las composiciones en el diagrama de clases navegacionales son interpretadas como la creación de un nodo de hipermedia compuesto, en la que varios nodos se muestran juntos.
- (2) El *diseño de las estructuras de navegación* establece las estructuras de acceso que permiten visitar los objetos del espacio navegacional. Están constituidas por menús, índices, visitas guiadas, y formularios. Todos ellos son clases con estereotipos. La notación de las estructuras de acceso es similar a la de RMM y se muestra en la figura 10.
 - ✓ Los **índices** tienen referencias a una colección de objetos, y permiten la navegación directa a ellos.
 - ✓ Las **visitas guiadas** contienen una colección de referencias, y permiten la navegación secuencial a través de la misma. Los índices y visitas guiadas pueden definir la colección de objetos a la que están asociados de forma dinámica mediante el uso de formularios de entrada y condiciones de selección. Por supuesto, los índices y visitas guiadas pueden referirse a colecciones fijas de objetos.
 - ✓ Un **menú** es un objeto navegacional que tiene un número fijo de asociaciones a estructuras de acceso u objetos.

- ✓ Un **formulario** permite al usuario ingresar información para completar las condiciones de selección de objetos pertenecientes a las colecciones de índices y visitas guiadas.
4. *Diseño de la presentación*: El modelo de presentación en UWE está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos de UML. Los elementos del modelo de presentación son: *anclas*, *entradas de texto*, imágenes, audio y botones. Cada clase del modelo navegacional tiene asignada una clase del modelo de presentación; las clases del modelo de presentación son equivalentes a las ADV de OOHDM.

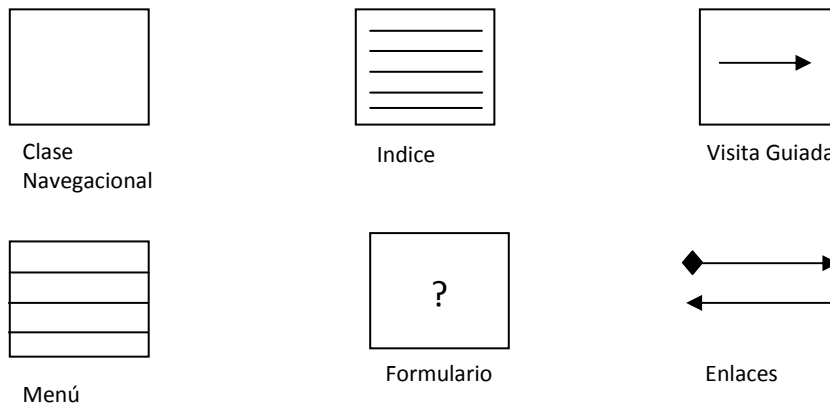


Figura 10 – Notación UWE

2.3.7.2. *Aplicación del método UWE al ejemplo*

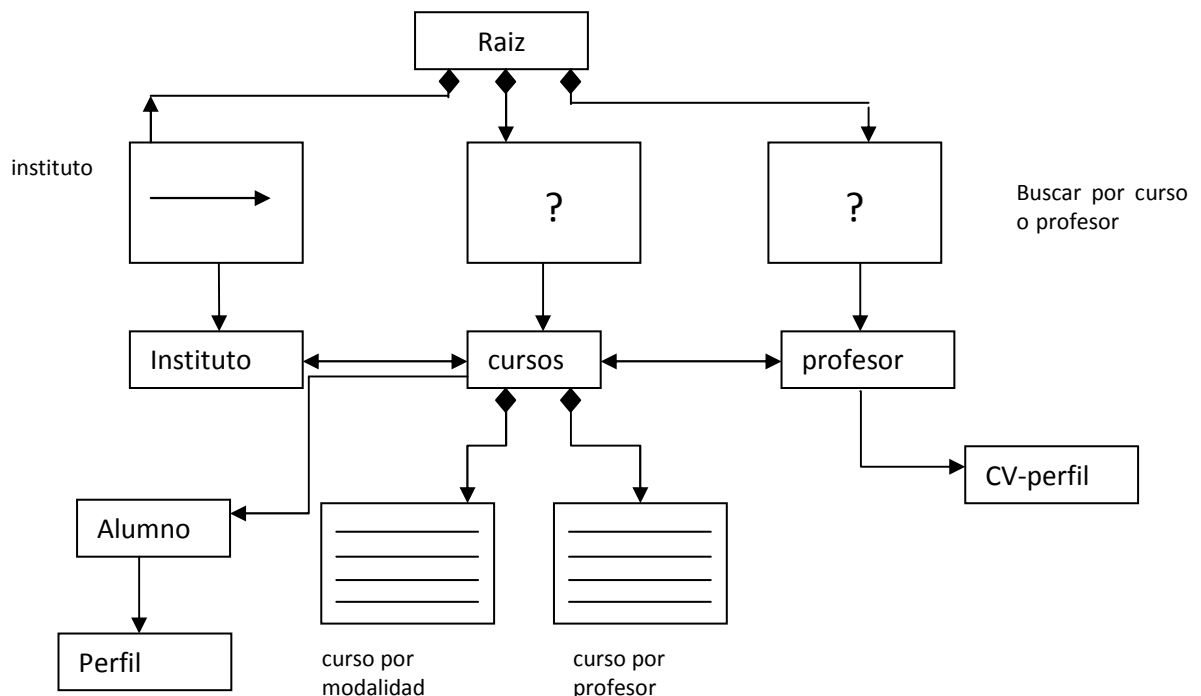


Figura 11 – Método UWE aplicado al ejemplo

2.4. Comparativa de Modelos

El método HDM fue el que estableció las etapas básicas en el proceso de generación de aplicaciones Web como: *análisis conceptual*, *diseño navegacional* y *presentación*, e *implementación*.

El *modelo conceptual* utiliza modelos relacionales u orientados a objetos, tal es el caso de RMM, OOHDH, UWE, OO-H, entre otros. Algunos métodos orientados a objetos utilizan la notación de UML.

El *modelo navegacional* captura las vistas de las entidades, reemplaza las relaciones 1-n y m-n por estructuras de acceso, y genera una página inicial con enlaces a estructuras de acceso que permiten la navegación a las entidades o clases del modelo navegacional. Cada método establece su propia notación para representar el modelo navegacional. Respecto de las notaciones se puede decir que la utilizada por el método OOHDH es la más robusta, aunque carece de un metamodelo bien definido que indique las construcciones válidas, de hecho se pueden encontrar variantes de su notación gráfica en cada publicación relacionada con el método. Las notaciones de UWE son más claras y están mejor documentadas. UWE tiene como ventaja adicional que es un *profile* de UML, es decir, un modelo navegacional es un diagrama de clases de UML con estereotipos.

En el *diseño de la presentación* casi todos los métodos siguen la técnica propuesta en OOHDH, que es el diseño abstracto de interfaces.

La fase de implementación no es abordada por los métodos, y en el mejor de los casos cuentan con alguna herramienta que permite generar código ejecutable en cierto lenguaje.

Los métodos OO-H y RMM poseen una notación clara.

El método que ha tenido más descendientes es OOHDH:

- ✓ SOHDH añade análisis por escenarios
- ✓ OO-H añade casos de uso
- ✓ UWE casos de uso y notación UML
- ✓ OOHDHDA una implementación mediante MDA.
- ✓ WSDM surge desde el punto de vista del diseño centrado en el usuario.

La siguiente, es una tabla comparativa de los métodos abordados en este trabajo:

Método	Proceso	Notación	Herramientas
HDM	Diseño a gran escala Diseño a pequeña escala	Nodos Enlaces Enlaces Navegacionales Elementos de Navegación	
RMM	Diseño E/R Diseño de Secciones Diseño Navegacional Protocolo de conversión Interfaz de usuario Comportamiento en tiempo de ejecución Construcción y pruebas	Entidad Sección Índice Visita guiada Enlaces bidireccional Enlace unidireccional Grupo Índice-visita guiada	RMDM
OOHDH	Diseño Conceptual Diseño Navegacional Diseño de Interfaz Abstracta Implementación	Diagrama de clases Diagrama de Contextos ADO ADV	OOHDH-Web HyperSD
OO-H	Análisis con casos de uso Modelado Conceptual Análisis y diseño navegacional Diseño de la presentación	Casos de uso Diagrama de clases Diagrama Navegacional OO-H	VisualWade

	Implementación	ADV	
OOWS	Modelo Conceptual: captura de requisitos, modelado conceptual, modelado de la navegación y de la presentación Desarrollo de la solución: Contextos navegacionales, enlaces navegacionales	Clases de uso Diagramas de clases Diagramas de contexto ADV	OOHDMDA
UWE	Análisis de requerimientos Diseño Conceptual Diseño Navegacional Diseño de la presentación Implementación y pruebas	Casos de uso Diagrama de clases Diagrama de clases con estereotipos Clases de presentación (ADV)	Argo-UWE

Tabla 2 – Comparativa de Modelos de diseño

2.4.1. Conclusión

Todos los métodos hacen hincapié en el diseño navegacional, como una etapa fundamental en el modelado de aplicaciones web. El diseño conceptual puede realizarse con herramientas clásicas de Ingeniería de Software, como modelado E/R o herramientas UML.

Capítulo 3 – Areas involucradas en el diseño de aplicaciones Web

3.1. Introducción

En el capítulo anterior se trataron diferentes metodologías de desarrollo de aplicaciones Web, como se puede apreciar ponen énfasis en el diseño navegacional. En el presente capítulo se tratará el tema desde el punto de vista del usuario final, que es a quien va dirigido el sistema.

El usuario final sólo desea *encontrar* la información (que busca) y que la misma sea de *calidad*, en esta frase, común en el mundo de los diseñadores de WebApps, se destacan dos palabras que serán tratadas en éste capítulo: “encontrar información” y “calidad de la información”.

El diseñador de una webApp debe tener especial cuidado cuando analiza hacia quién va dirigido el sistema, por ejemplo debe tener en cuenta que no es lo mismo diseñar un sitio web para una empresa que se encuentra en un centro urbano, que hacerlo para una escuela que se encuentra retirada de los centros urbanos o para un productor que se encuentra en el campo, en estos casos hay un punto importante a tener en cuenta: la velocidad de las conexiones. Un sitio web puede tener un diseño gráfico realmente artístico, pero en conexiones lentas el comportamiento del usuario, cansado de esperar respuestas a su solicitud, probablemente abandone el sitio sin evaluar si es bueno, si está bien diseñado, si cumple con las metodologías propuestas, simplemente al usuario no le sirve porque no da respuestas a su inquietud. Por tanto, también es preciso que el diseñador evalúe el *contexto*.

El buscador GOOGLE proporciona un buen ejemplo de síntesis, buen gusto, eficacia y eficiencia, es un sitio de *calidad* que ha logrado ganar el mercado, aunque el usuario ignore toda la tecnología subyacente.

A la hora de calificar la *calidad* de una aplicación web, un factor importante es la *usabilidad*, por ser el atributo visible y que determina el grado de satisfacción del usuario respecto de la aplicación web; de ello depende que sea utilizada o no.

3.2. Consideraciones previas - Planificación

La etapa de planificación de una aplicación Web debe tener en cuenta que: el rol del diseñador es el de ser una “interface” entre el usuario final (navegante) y el proveedor (dueño) del sitio.

Es en esta etapa donde se identifican los **objetivos del sitio**, se establecen necesidades, requerimientos y objetivos de los potenciales visitantes (navegantes).

Con esta información, se pueden definir los requerimientos en cuanto a recursos humanos y perfiles profesionales necesarios.

El **objetivo del sitio** debe ser la consecución de objetivos de proveedor y usuario para lo que el diseñador debe obtener información precisa de las necesidades y objetivos tanto del proveedor como del usuario. Para obtener esta información, debe realizar entrevistas y reuniones con los responsables del sitio por el lado del proveedor. En tanto, por el lado del usuario, debe establecer sus necesidades, objetivos, cómo se comporta y actúa, cuál será el contexto de uso y cómo afectará a la interacción su experiencia y conocimientos previos.

La planificación engloba métodos de aproximación contextual, estudios de campo o etnográficos, métodos de aproximación por grupos, métodos de aproximación individual (encuestas, cuestionarios y entrevistas). En esta etapa se recoge, analiza y ordena la información. Se trata de establecer una base sólida sobre la que se tomarán decisiones en etapas posteriores.

3.2.1. Consideraciones de Roger Pressman

Las WebApps, deben contener las siguientes características:

1. **Desarrollo basado en componentes:** Desarrollo de una infraestructura que permite la re-utilización de componentes probados con interacción a modo de sistema.

2. **Seguridad:** El uso de tecnología que no permita la intrusión, siendo ésta *la acción de penetrar de forma inadecuada o ilegítima en un lugar o una situación*.
3. **Cumplimiento de estándares:** El uso de lenguaje de marcas y metalenguaje como XML y HTML para describir una gran variedad de objetos y datos.

“El proceso de ingeniería comienza con la formulación que identifica las metas y objetivos de las WebApps y establece el ámbito del primer incremento”, establece el siguiente conjunto de actividades:

1. **Planificación:** Estima el coste global de proyecto, riesgos, planificación granulada para el incremento final y una menos granulada para los incrementos subsiguientes.
2. **Análisis:** Establece los requisitos técnicos e identifica los elementos del contenido que se van a incorporar. Toma en cuenta la estética.
3. **Ingeniería:** Incorporar el Diseño de contenido y la producción. Es el encargo del video, sonido y texto de la WebApp para personas no técnicas. Se llevan a cabo tareas de diseño.
4. **Generación de páginas:** Se produce la fusión de contenido y el diseño arquitectónico. También se lleva a cabo la integración de Middleware o componentes. Las pruebas buscan encontrar errores en los componentes, y aseguran el funcionamiento del WebApp.

Cada incremento obedecerá la aceptación del cliente (dueño del sitio). Estos cambios proporcionan el punto de partida para un nuevo ciclo de desarrollo.

Según Powell^{xiii}, las preguntas de inicio sugeridas son:

- ✓ ¿Cuál es la motivación principal para la WebApp?
- ✓ ¿Por qué es necesaria la WebApp?
- ✓ ¿Quién va a utilizar la WebApp?
- ✓ Al final, el cliente define la demografía para la Web.
- ✓ Indicar el grado de integración con otros sistemas, como por ejemplo, bases de datos.
- ✓ Preguntar por los temas de conectividad.

En cuanto al análisis, detalla las siguientes actividades:

- ✓ **Análisis de contenido:** Identificar el contenido que se va a proporcionar.
- ✓ **Análisis de interacción:** Se detalla la interacción del usuario y la WebApp
- ✓ **Análisis funcional:** Casos de uso que definen las operaciones en el contenido y operaciones de procesamiento.
- ✓ **Análisis de configuración:** Descripción del entorno e infraestructura del WebApp

El producto de la actividad de análisis debe documentarse adecuadamente para tener una base respecto al incremento del siguiente desarrollo.

En cuanto al diseño, Pressman establece que se debe tomar en cuenta cuatro elementos técnicos:

- ✓ **Principios y métodos de diseño:** Modularidad, elaboración paso a paso y cualquier otra heurística de diseño de software. Uso de hipermedia (interacción de objetos con un protocolo)
- ✓ **Reglas de oro:** Reutilizar el conjunto de heurística de diseño.
- ✓ **Configuración de diseño:** Aplicar el enfoque genérico que resuelve pequeños problemas para aplicar la solución a una variedad más amplia.
- ✓ **Plantillas:** Marco de trabajo esquemático de una configuración de diseño o documento.

El enfoque de pruebas será el mismo que se haya recomendado para las pruebas de sistemas orientados a objetos:

- ✓ El modelo de contenido es revisado para describir errores.
- ✓ El modelo de diseño es revisado para descubrir errores de navegación.

- ✓ Se aplican pruebas de unidad a los componentes de proceso seleccionados y las páginas Web.
- ✓ Se construye la arquitectura y se realizan las pruebas de integración.
- ✓ La WebApp ensamblada se prueba para conseguir una funcionalidad global.
- ✓ Se implementa en una variedad de configuraciones diferentes de entornos y comprobar así la compatibilidad con cada configuración.
- ✓ Se prueba con una población de usuarios finales controlada y monitorizada.

Se debe tener una planificación en la que se consideren los riesgos, establecer planes temporales y definir controles. No se debe dejar al WebApp sin consideración post-instalación.

- ✓ Deben conformarse los siguientes equipos:
 - Desarrolladores y proveedores de contenido
 - Editores de Web: Intermedio entre Personal no técnico y encargados técnicos.
 - Ingeniero de Web, encargado del proceso y la tecnología.
 - Especialista de soporte: Continuidad de soporte.
 - Administrador o Webmaster: responsable del funcionamiento diario del Web.

3.3. Calidad

La calidad de un producto o servicio web es la percepción que el navegante tiene del mismo, es una fijación mental del consumidor que asume conformidad con dicho producto o servicio y la capacidad del mismo para satisfacer sus necesidades. Por tanto, debe definirse en el contexto que se esté considerando.

Desde el punto de vista que se está abordando, entregar una WebApp de calidad, significa aportar valor al usuario, ofrecer condiciones de uso del servicio superior a las que el usuario espera recibir y a un precio accesible.

En una visión actual, el concepto de calidad es entregar al cliente no lo que quiere, sino lo que nunca se había imaginado que quería y que una vez que lo obtenga, se dé cuenta que era lo que siempre había querido.

Para la Real Academia Española (RAE), calidad es: *“Propiedad o conjunto de propiedades inherentes a una cosa que permiten apreciarla como igual, mejor o peor que las restantes de su especie”*

La norma ISO 8402 define la calidad como: *“el conjunto de propiedades y características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas”*.

La norma ISO 9000 dice de calidad: *“Calidad: grado en el que un conjunto de características inherentes cumple con los requisitos”*

3.3.1. Otras definiciones de calidad

- ✓ Luis Andres Arnauda sequera: Define la norma ISO 9000 "Conjunto de normas y directrices de calidad que se deben llevar a cabo en un proceso".
- ✓ Philip Crosby: "Calidad es cumplimiento de requisitos"
- ✓ Joseph Juran: "Calidad es adecuación al uso del cliente".
- ✓ Armand V. Feigenbaum: "Satisfacción de las expectativas del cliente".
- ✓ Genichi Taguchi: "Calidad es la menor pérdida posible para la sociedad".
- ✓ William Edwards Deming: "Calidad es satisfacción del cliente".
- ✓ Walter A. Shewhart: "La calidad como resultado de la interacción de dos dimensiones: dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece)".

3.3.2. Cómo se mide la calidad de una WebApp?

Como se ha venido analizando, la calidad de una WeApp contiene factores que pueden ser medidos:

- ✓ Eficiencia
- ✓ Flexibilidad
- ✓ Facilidad de uso
- ✓ Portabilidad
- ✓ Seguridad, etc.

3.3.3. Modelo de Mc Call

La ISO 9126 de 2001, basada en el modelo de Mc Call, plantea un modelo normalizado que permite evaluar y comparar productos sobre la misma base.

En este modelo, la calidad queda definida mediante seis características:

1. **Funcionalidad:** Las funciones satisfacen necesidades declaradas o implícitas
2. **Fiabilidad:** Capacidad de un sistema para mantener su nivel de rendimiento
3. **Usabilidad:** Esfuerzo necesario para el uso y la valoración individual de tal uso, por parte de un conjunto de usuarios.
4. **Portabilidad:** Es la capacidad de un sistema para ser transferido de un entorno a otro.
5. **Mantenibilidad:** Es el esfuerzo necesario para realizar modificaciones específicas.
6. **Eficiencia:** Es la relación entre el nivel de prestaciones de un sistema y el volúmen de recursos utilizados en condiciones declaradas.

Este estándar no proporciona métricas ni métodos de medición, por lo que no son prácticas las mediciones directas de las características de calidad.

Para resolver este problema, González (2009), revisó la ISO 9126 e incluyó un nuevo modelo de calidad que distingue entre tres aproximaciones a la calidad de producto en ISO 14598, que son:

1. **Calidad Interna:** Se mide por las propiedades estáticas del código, utilizando técnicas de inspección
2. **Calidad externa:** Se mide por las propiedades dinámicas del código cuando éste se ejecuta
3. **Calidad en uso:** Se mide por el grado por el cual el software está realizado en función de las necesidades del usuario en el entorno de trabajo para el que fue construido

3.3.4. Modelo de Olsina

Luis Olsina establece una lista para caracterizar a las WebApps de alta calidad:

1. **Usabilidad**
 - (a) Capacidad de comprensión del sitio global
 - (b) Servicio de ayuda y realimentación en línea
 - (c) Capacidad estética y de interface.
 - (d) Servicios especiales.
2. **Funcionalidad**
 - (a) Capacidad de recuperación y de búsqueda
 - (b) Servicios de búsqueda y navegación
 - (c) Servicios relacionados con el dominio de la aplicación
3. **Fiabilidad**
 - (a) Proceso correcto de enlace
 - (b) Recuperación de errores
 - (c) Validación y recuperación de entrada del usuario
4. **Eficiencia**
 - (a) Rendimiento del tiempo de respuesta
 - (b) Velocidad de generación de páginas
 - (c) Velocidad de generación de gráficos
5. **Capacidad de mantenimiento**
 - (a) Facilidad de corrección

- (b) Adaptibilidad
- (c) Extensibilidad

3.4. Usabilidad

En la etapa de diseño se toman decisiones acerca de cómo diseñar o rediseñar en base al conocimiento obtenido en la etapa de Planificación, teniendo en cuenta que la usabilidad es un factor estratégico para conseguir un máximo aprovechamiento de los recursos proporcionados por la web.

Debe ser entendida siempre en relación con las características y necesidades y con la forma y condiciones de uso por parte de sus navegantes (usuario final).

Un diseño no es en sí mismo usable: " *lo es para usuarios específicos en contextos de uso específico*".

La usabilidad se compone de dos tipos de atributos:

- ✓ *Atributos cuantificables de forma objetiva*: eficacia o número de errores cometidos por el usuario durante la realización de una tarea, y eficiencia o tiempo empleado por el usuario para la consecución de una tarea.
- ✓ *Atributos cuantificables de forma subjetiva*: satisfacción de uso, medible a través de la interrogación al usuario.

La ingeniería de software se ocupa de aquellos atributos del software como fiabilidad, seguridad, productividad, relacionados con características internas del sistema, sin tener en cuenta la perspectiva del usuario, el caso de las WebApp es diferente porque deben ser diseñadas para que sean usadas por los usuarios-navegantes (interacción entre los usuarios y el sistema web).

El término *usabilidad* según Bevan, Kirakowsky, Maissel, es planteado desde tres puntos de vista:

1. **Orientado al producto**
2. **Orientado al usuario**
3. **Orientado al rendimiento del usuario**

La Organización Internacional para la Estandarización (ISO) ofrece dos definiciones de usabilidad:

1. ISO/IEC 9126: "La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso". Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su funcionalidad y eficiencia. La usabilidad depende no sólo del producto sino también del usuario. Por ello un producto no es en ningún caso intrínsecamente usable, sólo tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares. La usabilidad no puede ser valorada estudiando un producto de manera aislada (Bevan, 1994).
2. ISO/IEC 9241: "Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico". Es una definición centrada en el concepto de calidad en el uso, es decir, se refiere a cómo el usuario realiza tareas específicas en escenarios específicos con efectividad.

A partir de la conceptualización llevada a cabo por la ISO, se infieren los principios básicos en los que se basa la usabilidad:

1. **Facilidad de Aprendizaje**: facilidad con la que nuevos usuarios-navegantes desarrollan una interacción efectiva con el sistema o producto. Está relacionada con la previsibilidad, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.
2. **Flexibilidad**: relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la

multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.

3. **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.

En informática, la usabilidad está muy relacionada con la accesibilidad, hasta el punto de que algunos expertos consideran que una forma parte de la otra o viceversa. Uno de estos expertos de la usabilidad en los entornos web es Jakob Nielsen, quien definió la usabilidad en el 2003 como "*un atributo de calidad que mide lo fáciles de usar que son las interfaces web*".

Otra definición clarificadora es la de Redish, para quien es preciso diseñar sitios web para que los usuarios sean capaces de "*encontrar lo que necesitan, entender lo que encuentran y actuar apropiadamente... dentro del tiempo y esfuerzo que ellos consideran adecuado para esa tarea*".

La ergonomía parte de los principios del diseño universal o diseño para todos. El diseñador de ergonomía proporciona un punto de vista independiente de las metas de la programación porque el papel del diseñador es actuar como defensor del usuario. Por ejemplo, tras interactuar con los usuarios, el diseñador de ergonomía puede identificar necesidades funcionales o errores de diseño que no hayan sido anticipados.

La ergonomía incluye consideraciones como:

- ✓ ¿Quiénes son los usuarios, cuáles sus conocimientos, y qué pueden aprender?
- ✓ ¿Qué quieren o necesitan hacer los usuarios?
- ✓ ¿Cuál es la formación general de los usuarios?
- ✓ ¿Cuál es el contexto en el que el usuario está trabajando?
- ✓ ¿Qué debe dejarse a la máquina? ¿Qué al usuario?

Las respuestas a estas preguntas pueden conseguirse realizando análisis de usuarios y tareas al principio del proyecto.

Otras consideraciones incluyen:

- ✓ ¿Pueden los usuarios realizar fácilmente sus tareas previstas? Por ejemplo, ¿pueden los usuarios realizar las tareas previstas a la velocidad esperada?
- ✓ ¿Cuánta preparación necesitan los usuarios?
- ✓ ¿Qué documentación u otro material de apoyo están disponible para ayudar al usuario? ¿Puede éste hallar las respuestas que buscan en estos medios?
- ✓ ¿Cuáles y cuántos errores cometen los usuarios cuando interactúan con el producto?
- ✓ ¿Puede el usuario recuperarse de los errores? ¿Qué han de hacer los usuarios para recuperarse de los errores? ¿Ayuda el producto a los usuarios a recuperarse de los errores? Por ejemplo, ¿muestra el software mensajes de errores informativos y no amenazantes?
- ✓ ¿Se han tomado medidas para cubrir las necesidades especiales de los usuarios con discapacidades? (Es decir, ¿se ha tenido en cuenta la accesibilidad?)

Las técnicas para encontrar respuesta a estas y otras cuestiones son: (1) análisis de requisitos enfocado al usuario, (2) construcción de perfiles de usuarios y (3) pruebas de usabilidad.

Para Jakob Nielsen, el principal problema de la usabilidad reside en que es una cualidad demasiado abstracta para ser medida directamente. Por esta razón la descompone en cinco atributos para poder estudiarla:

1. Facilidad de aprendizaje
2. Eficiencia
3. Recuerdo en el tiempo
4. Tasa de errores
5. Satisfacción.

Métodos de evaluación de usabilidad: Considera que un sitio web usable es aquél en el que los usuarios pueden interactuar de la forma más fácil, cómoda, segura e inteligentemente posible.

No sólo la tecnología y el aspecto gráfico son factores determinantes para hacer un sitio web llamativo. Es importante que cumpla con las siguientes características:

- ✓ Entendible
- ✓ Novedoso
- ✓ Comprensible
- ✓ Inteligente
- ✓ Atractivo

En síntesis, la finalidad de un sitio Web es lograr que el usuario-navegante encuentre lo que busca en el menor tiempo posible. La Usabilidad de un sitio web está determinada por sus contenidos, entre más cercanos estén al usuario, mejor es la navegación por el mismo y más acertada será la experiencia al enfrentarse a la pantalla.

Los diseñadores y creadores deben tratar de mostrar todos los elementos de una manera clara y concisa.

En ocasiones los cibernautas se enfrentan a sitios web de altísima calidad y contenido, pero que presentan dificultades en su contenido. Por ejemplo, que los menús son de difícil ubicación, o que la herramienta de búsqueda no aparece en un lugar visible.

Aunque no hay estándares definidos para la Usabilidad, depende en cierta forma del espacio donde se desenvuelve el navegante. Pero lo importante en este caso es que el usuario no se deje consumir ni dominar por el sitio, es decir que sea él mismo que tome el control de la navegación por medio de un aprendizaje sencillo y el dominio de los elementos necesarios, para encontrar finalmente y en el menor tiempo posible, lo que busca. Un buen sitio Web debe responder a las necesidades del usuario. En una comunidad virtual donde confluyen diferentes culturas e intereses, el contexto en el que se desenvuelven los miembros de un grupo virtual, o comunidad, no puede generar molestias en el momento de la navegación.

Un error recurrente de los creadores y diseñadores de sitios Web, es querer imponer sus decisiones y criterios sin pensar en el usuario-navegante. Por eso en el momento de diseñar el sitio e introducir contenidos, siendo esta última labor de los editores, y no de los diseñadores, es importante pensar en el otro.

3.4.1. Beneficios de la Usabilidad

Entre los principales beneficios encontramos:

1. Reducción de los costes de aprendizaje.
2. Disminución de los costes de asistencia y ayuda al usuario.
3. Optimización de los costes de diseño, rediseño y mantenimiento.
4. Aumento de la tasa de conversión de visitantes a clientes de un sitio web.
5. Mejora la imagen y el prestigio.
6. Mejora la calidad de vida de los usuarios, ya que reduce su estrés, incrementa la satisfacción y la productividad.

Todos estos beneficios implican una reducción y optimización general de los costes de producción, así como un aumento en la productividad. La usabilidad permite mayor rapidez en la realización de tareas y reduce las pérdidas de tiempo.

Después de ser rediseñado, prestándose especial atención a la usabilidad, el sitio web de IBM incrementó sus ventas en un 400% (InfoWorld, 1999).

3.4.2. UEM [Usability Evaluation Methods]

Se pueden considerar dos grupos de UEM (Métodos de Evaluación de la Usabilidad):

1. Los UEM empíricos en el que se utilizan técnicas tales como pensar en voz alta, participan:

- a. Usuarios
 - b. Evaluadores
 - c. Observadores
 - d. Expertos en test
2. Los UEM analíticos donde no tienen acceso los usuarios-navegantes: incluye un equipo de especialistas en usabilidad. Para el proceso de inspección se utilizan directrices o heurísticas del proceso de inspección.

3.4.3. Métricas de usabilidad

Por medición se entiende el proceso de atribuir números o símbolos a los *atributos de las entidades en el mundo real*, una métrica es la correspondencia del mundo real a un mundo formal, a través de la medición es posible *juzgar lo que se mide*.

Una métrica es un valor numérico asignado a *entidades del mundo real*: software, sitio web, aplicación web, etc.

Un *atributo* es la característica de una entidad de tipo directo o indirecto, por ejemplo: links no operativos, microcódigo no accesible, etc.

El uso de métricas no limita la intervención humana y ofrece una reducción de la subjetividad en la evaluación de calidad de un sitio o aplicación web, etc.

3.4.4. Métricas de evaluación de calidad

Estas son las métricas propuestas en el estándar ISO 9126-4:

- ✓ Métricas relacionadas con la efectividad
- ✓ Métricas relacionadas con la productividad
- ✓ Métricas relacionadas con la seguridad
- ✓ Métricas relacionadas con la satisfacción

La usabilidad de la aplicación depende no sólo del diseño del interfaz, sino también de su arquitectura o sea de su estructura y organización, en otras palabras, del componente no visible del diseño. El diseño a nivel de arquitectura tiene una gran influencia en la usabilidad del sistema

3.5. Arquitectura de la Información (AI)

La Arquitectura de la Información, es definida como el arte y la ciencia de organizar espacios de información con el fin de ayudar a los usuarios a satisfacer sus necesidades de información. *La actividad de organizar comporta la estructuración, clasificación y rotulado de los contenidos del sitio web* (Toub; 2000) .

Hay dos aspectos a tener en cuenta cuando se diseña la Arquitectura de la Información:

1. Que el usuario-navigate ENCUENTRE la información
2. El diseño a nivel conceptual: Las técnicas propias de la AI, dentro del ciclo de vida del desarrollo del sitio, se ubican en fases de diseño conceptual. Las fases de diseño visual están, en cambio, envueltas por técnicas de Ingeniería de la Usabilidad, Diseño de Interfaces y Diseño de Información

Pressman define la estructura de la hipermedia, configuración de diseño y plantillas. El diseño de contenido es una actividad no técnica que involucra la estructura y formato de la información a presentar como parte de la WebApp.

Los patrones de diseño citados son:

- ✓ **Ciclo:** Devuelve al usuario al nodo de contenido anterior.
- ✓ **Anillo de Web:** Un ciclo que enlaza hipertextos hablando de un tema.
- ✓ **Contorno:** Varios ciclos inciden en uno.
- ✓ **Contrapunto:** Comentarios de hipertexto interrumpiendo la narrativa del contenido.

- ✓ **Mundo de espejo:** Uso de diferentes hilos narrativos.
- ✓ **Tamiz:** Presenta conjunto de opciones de decisión para el usuario.
- ✓ **Vecindario:** Marco de navegación uniforme por todas las páginas Web.

Según Pressman :

- (1) el diseñador del WebApp crea una unidad semántica de navegación de acuerdo a los roles de los usuario. Es importante el uso de re-alimentación e iconografía apropiada.
- (2) La interfaz del WebApp es la primera impresión para el usuario. Se debe considerar los siguientes criterios de diseño:
 - ✓ Probabilidad de errores del servidor.
 - ✓ Cantidad de lectura de texto.
 - ✓ Evitar símbolo bajo construcción.
 - ✓ Evitar el desplazamiento en la pantalla
 - ✓ Diseño consecuente de menús de navegación y barra de cabecera.
 - ✓ Equilibrio entre estética y funcionalidad.
 - ✓ Opciones obvias de navegación.

Capítulo 4 – Del diseño Instruccional

4.1. Introducción

Este capítulo introduce algunos conceptos necesarios que se deben tener en cuenta a la hora de diseñar herramientas hipermediales para ser usadas en el proceso enseñanza-aprendizaje. Aprendizaje Significativo, Diseño Instruccional, Teorías de Aprendizaje, son conceptos que no son tenidos en cuenta por los ingenieros Web a la hora de desarrollar una herramienta educativa. Sin duda, no es mala intención, simplemente porque su carrera no los preparó para enseñar, sino para desarrollar herramientas. En general, el ingeniero toma como modelo la enseñanza dejada por sus propios profesores, más un toque de curiosidad, improvisación y sentido común.

En esta nueva era a la que se le dio en llamar “era del conocimiento”, el sector más afectado es el educativo porque influye en la formación del capital intelectual. El modelo educativo debe reunir a la comunidad académica en espacios físicos o virtuales usando las Nuevas Tecnologías de Información y Comunicación (NTICs) como un complemento socializador. En este escenario, Internet, pasa a ser una herramienta que se amolda a este nuevo modelo educativo.

La simple observación de las conductas de los jóvenes estudiantes nos muestra claramente cómo las tecnologías digitales impactan sobre los estilos de aprendizajes de los alumnos y por ende, las estrategias de enseñanza de los docentes están obligadas a cambiar. Las nuevas tendencias a nivel mundial sugieren reflexionar sobre lo que se enseña, cómo se enseña y cómo se evalúan los aprendizajes en los que intervienen estas tecnologías.

Siempre es importante tener en cuenta el contexto en el que se desarrollan las nuevas generaciones, para los jóvenes las nuevas tecnologías son “su modo de ver al mundo”. Según Roxana Morduchowicz los jóvenes de hoy suelen definirse a sí mismos por su relación con la cultura popular, entendida como aquella que construyen los medios de comunicación, la música, el cine y otras expresiones culturales. Los jóvenes no se levantan con el despertador sino con el celular y el ipod. Es común ver a los jóvenes “conectados” a dispositivos de audio. Sus vidas son contadas en páginas Web y fotologs.

A esta generación se la que se denomina “generación M” o “generación multimedia”, también se los conoce como “Learning by doing”: aprender haciendo. Estos adolescentes, tienen una gran adaptabilidad tecnológica y habitualmente pueden trasladar su capacidad de una tecnología a otra con absoluta naturalidad^{xiv, xv}.

Por tanto el docente debe aprovechar y conquistar ese potencial para mejorar el proceso enseñanza-aprendizaje. El diseño instruccional debe enmarcarse en el modelo constructivista por el tipo de lenguaje educativo-comunicacional, por la estructuración de la información y el grado de libertad que se debe otorgar al aprendiz.

“El conocimiento no está en el contenido disciplinar, sino en la actividad constructiva (o co-constructiva) de la persona sobre el dominio de contenido tal como ocurre en un contexto socioeducativo determinado” (Jerome Bruner)

El rol del agente educativo es el de modelar, esto es: promover el descubrimiento individual y colaborativo para solucionar problemas reales.

4.2. Aprendizaje significativo

“El analfabeto de mañana, no será la persona incapaz de leer. El analfabeto de mañana será la persona que no haya aprendido a aprender” (Alvin Toffler)

4.2.1. Aprendizaje significativo:

Es el aprendizaje en el cual el sujeto relaciona sus conocimientos previos con los nuevos dotándolos así de coherencia respecto a sus estructuras cognitivas.(Ausubel)

En 1963, David Paul Ausubel hizo su primer intento de explicación de una teoría cognitiva del aprendizaje verbal significativo en la monografía “The Psychology of Meaningful Verbal Learning”; en el mismo año se celebró en Illinois el Congreso Phi, Delta, Kappa, en el que intervino con la ponencia “Algunos aspectos psicológicos de la estructura del conocimiento”.

En la década de los 70's, Jerome Bruner propuso el Aprendizaje por Descubrimiento, afirmando que el aprendiz construye su conocimiento a través del descubrimiento de contenidos, por lo que se puso énfasis al activismo y los experimentos dentro del aula, criticándose el modelo expositivo tradicional.

David Paul Ausubel reconoció las bondades del aprendizaje por descubrimiento, pero se opuso a su aplicación irreflexiva, considera que tiene una desventaja: necesita más tiempo para la realización de actividades.

El aprendizaje por descubrimiento no debe considerarse como opuesto al aprendizaje por recepción (aprendizaje que resulta de una exposición), pues la exposición puede ser de calidad (eficaz), si se dan ciertas características. Por otro lado puede ser más eficiente, porque se invierte menos tiempo.

En el caso del Aprendizaje Memorístico, los nuevos conocimientos se incorporan de forma arbitraria en la estructura cognitiva del alumno, quien no realiza ningún esfuerzo para integrar los nuevos conocimientos con sus conocimientos previos, tampoco quiere aprender, porque no concede valor a los contenidos presentados por el profesor.

Ausubel considera aprendizajes de baja calidad a los memorísticos o repetitivos, en cambio, el aprendizaje significativo puede darse por recepción o por descubrimiento, y puede lograr en el alumno aprendizajes de calidad.

El aprendizaje por recepción no implica, como mucho se critica, una actitud pasiva del alumno; ni tampoco las actividades diseñadas para guiar el aprendizaje por descubrimiento garantizan la actividad cognoscitiva del alumno.

Las características del Aprendizaje Significativo, según Ausubel, son:

- ✓ Los nuevos conocimientos se incorporan en forma sustantiva en la estructura cognitiva del alumno.
- ✓ Esto se logra gracias a un esfuerzo deliberado del alumno por relacionar los nuevos conocimientos con sus conocimientos previos.
- ✓ Todo lo anterior es producto de una implicación afectiva del alumno, es decir, el alumno *quiere aprender* aquello que se le presenta porque lo considera valioso.

4.2.1.1. Ventajas del Aprendizaje Significativo

El Aprendizaje Significativo tiene claras ventajas sobre el Aprendizaje Memorístico:

- ✓ Produce una retención más duradera de la información. Modificando la estructura cognitiva del alumno mediante reacomodos de la misma para integrar a la nueva información.
- ✓ Facilita el adquirir nuevos conocimientos relacionados con los ya aprendidos en forma significativa, ya que al estar claramente presentes en la estructura cognitiva, se facilita su relación con los nuevos contenidos.
- ✓ La nueva información, al relacionarse con la anterior, es depositada en la llamada memoria a largo plazo, en la que se conserva más allá del olvido de detalles secundarios concretos.
- ✓ Es activo, pues depende de la asimilación deliberada de las actividades de aprendizaje por parte del alumno.
- ✓ Es personal, pues la significación de los aprendizajes depende de los recursos cognitivos del alumno (conocimientos previos y la forma como éstos se organizan en la estructura cognitiva).

De acuerdo a la teoría de Ausubel, para que se produzca aprendizaje significativo deben darse dos condiciones fundamentales:

1. *Actitud potencialmente significativa de aprendizaje por parte del aprendiz*, o sea, predisposición para aprender de manera significativa.
2. Presentación de un material potencialmente significativo. Esto requiere:
 - *Que el material tenga significado lógico*: que sea potencialmente relacionable con la estructura cognitiva del que aprende de manera no arbitraria y sustantiva.
 - *Que existan ideas de anclaje o subsumidores adecuados en el sujeto* que permitan la interacción con el material nuevo que se presenta.

En cuanto a la *actitud potencialmente significativa por parte del aprendiz* no basta que el que el alumno quiera aprender para que se dé el aprendizaje significativo, también es necesario que *pueda* aprender, que entienda el material. Aunque, si el alumno no quiere aprender, el aprendizaje no puede producirse. El aprendizaje significativo es un componente de disposiciones emocionales y actitudinales, en el que el profesor puede influir a través de la *motivación*.

Al decir que el *material tenga significado lógico*, significa que el material presentado tenga una estructura interna organizada, que sea susceptible de dar lugar a la construcción de significados. *No sólo es importante el contenido, sino también la forma en que es presentado*.

Cuando establece que *existan ideas anclaje o subsumidores adecuados*, se refiere a la posibilidad de que el alumno conecte el conocimiento presentado con los conocimientos previos, ya incluidos en su estructura cognitiva. Los contenidos entonces son comprensibles para el alumno. El alumno debe contener ideas inclusoras en su estructura cognitiva, si esto no es así, el alumno guardará en memoria a corto plazo la información para contestar un examen memorista, y olvidará después, y para siempre, ese contenido.

Cuando un profesor (adulto) ha asimilado un contenido, a veces olvida que esto es un proceso que para el alumno representa un esfuerzo de *acomodación* de su estructura cognitiva.

El aprendizaje de proposiciones es el que se apoya mediante el uso adecuado de mapas conceptuales, que permiten visualizar los procesos de asimilación de los alumnos respecto a los contenidos que se pretende que aprendan, de forma de poder identificar oportunamente posibles errores u omisiones e intervenir para corregirlos.

Del conocimiento de los requisitos para que un aprendizaje se dé en forma significativa, se desprenden consecuencias de tipo didáctico.

Es necesario estar al tanto de los conocimientos previos del alumno, que el contenido a presentar pueda relacionarse con ideas previas, por lo que el conocer qué saben los alumnos sobre el tema ayudará a intervenir sobre el plan de la currícula. El mismo Ausubel escribe, como frase introductoria de su clásico libro *Psicología Educativa*: "*Si tuviese que reducir toda la psicología educativa a un solo principio, enunciaría éste: el factor más importante que influye en el aprendizaje es lo que el alumno ya sabe. Averígüese esto, y enséñese en consecuencia*".

Otro punto importante a abordar, es la organización del material del curso. Debe tener forma lógica y jerárquica, no se debe perder de vista que no sólo es importante el contenido sino la forma en que éste es presentado a los alumnos, por lo que se deberá presentar en secuencias ordenadas, de acuerdo a su potencialidad de inclusión.

Por último, considerar la importancia de la *motivación* del alumno. Recordar que si el alumno no quiere, no aprende. De modo que se debe dar motivo para *querer aprender* aquello que se le presenta. El que el alumno tenga entonces una actitud favorable, el que se sienta contento en la clase, el que estime a su profesor, no son románticas idealizaciones del trabajo en el aula sino que deberán buscarse intencionalmente por quienes se dedican profesionalmente a la educación. Como afirma Don Pablo Latapí: "*si tuviera que señalar un indicador y sólo uno de la calidad en nuestras escuelas, escogería éste: que los alumnos se sientan a gusto en la escuela*".

La Teoría del Aprendizaje Significativo tiene importantes implicaciones psicológicas y pedagógicas. Considera que el aprendizaje se construye de manera evolutiva porque se ocupa de lo que ocurre en el aula, postula los principios programáticos para organizar la docencia y, en este sentido, adquiere un valor especial la necesidad de realizar un análisis conceptual del contenido que huya de planteamientos simplistas. Es una teoría viva que no sólo se ha limitado a resistir durante tanto tiempo, sino que ha evolucionado a lo largo de su historia, a través de las distintas contribuciones que ha recibido. La aplicación de sus principios a la investigación en educación y a la enseñanza ha permitido, no sólo validar su conocimiento, sino también ampliarlo con interesantes aportes que han enriquecido su aplicación y su potencialidad explicativa.

Los constructos de modelo mental y esquema de asimilación permiten explicar el proceso de construcción del aprendizaje significativo y, por tanto, la adquisición, la asimilación y la retención del conocimiento.

Para Frida Díaz Barriga, “la función del trabajo docente no puede reducirse ni a la de simple transmisor de la información, ni a la de facilitador del aprendizaje. Antes bien, el docente se constituye en un *mediador* en el encuentro del alumno con el conocimiento. En esta mediación el profesor orienta y guía la actividad mental constructiva de sus alumnos, a quienes proporciona ayuda pedagógica ajustada a su competencia”.

La UNESCO propone: “nuestros alumnos deberán aprender a *conocer, a hacer, a ser y a convivir*. Utilicemos, pues, técnicas y estrategias que propicien todos estos aprendizajes. En esta perspectiva, afirmamos lo que con el pasar de las páginas será evidente, los mapas conceptuales son una herramienta útil para propiciar aprendizajes significativos en estos cuatro pilares”.

4.3. Del Diseño Instruccional

Diseño Instruccional (DI) es el proceso que funciona de manera continua y sistemática que genera la prevención de especificaciones instruccionales por medio del uso de teorías instruccionales y teorías de aprendizaje para asegurar que se alcanzarán los objetivos planteados. En el diseño instruccional se hace un completo análisis de las necesidades y metas educativas a cumplir y, posteriormente, se diseña e implementa un mecanismo que permita alcanzar esos objetivos. Así, este proceso involucra el desarrollo de materiales y actividades instruccionales, luego las pruebas y evaluaciones de las actividades del alumno.

El diseño instruccional implica una "*planeación*" sobre el curso que se desea impartir (qué se pretende, a quién estará dirigido, qué recursos y actividades serán necesarios, cómo se evaluará y cómo se mejorará), para lo cual se requiere organizar información y definir objetivos de aprendizaje: claros y medibles.

El diseñador instruccional debe pensar qué métodos, estrategias, actividades y recursos deberá utilizar para que los estudiantes aprendan y den sentido a la información que recibirán.

Tipos de objetivos del diseño instruccional:

- ✓ **Generales:** es un objetivo a largo plazo, expresa conductas últimas y conocimientos mínimos de la persona, deben plantearse con verbos de conductas no observables.
- ✓ **Particulares:** se considera un objetivo a mediano plazo, expresan conductas básicas, los verbos con los cuales se deben formular deben ser de conductas observables.
- ✓ **Específicos:** expresan conductas observables, manifiestan que el individuo ha logrado un aprendizaje, y los verbos para formularlos siempre son de conductas observables.

4.3.1. Teorías del Diseño Instruccional

Los Nueve Pasos de Gagné:

1. Atraer Atención
2. Informar Objetivo de Aprendizaje

3. Evocar Conocimientos Previos
4. Presentar Materiales
5. Guiar proceso de aprendizaje
6. Experimentar o puesta en práctica
7. Proveer Retroalimentación
8. Evaluar Transferencia o Rendimiento
9. Material Adicional: Ejercicios, ejemplos, casos, repaso, etc.

Modelo ARC de Keller: Componentes del Modelo Motivacional

1. **Atención:** Interés (perceptual arousal): duda o elemento sorpresa atraer atención. Curiosidad (inquiry arousal): curiosidad y novedad motivar preguntas. Métodos: ejemplos, dinámicas, debate, humor, multi-medios, multi-estilos.
2. **Relevancia:** Beneficios, actualidad, opciones para temas o destrezas adquiridas.
3. **Confidencia:** Guía superar reto: objetivos, pasos, tips, orientación, etc.
4. **Satisfacción:** Haber estado motivado y haber logrado superar el reto.

Componentes de Merrill: Component Display Theory [Clark00]

1. **Aprendizaje:** De Contenido: Hechos, conceptos, procedimientos y principios. De Desempeño: Memorizar, aplicar y deducir nuevas abstracciones.
2. **Tipos de Presentación:** Reglas. Ejemplos. Recordar eventos. Prácticas.

Teoría de Elaboración de Reigeluth:

De lo simple a lo complejo y de lo general o lo específico siguiendo los 7 pasos para la elaboración:

1. Secuenciación.
2. Organización.
3. Sumarización.
4. Síntesis.
5. Analogía.
6. Activador Estrategia de Cognición.
7. Control.

ISD (Instructional System Design) ó SAT (System Approach to Training):

Modelo ADDIE - Análisis, Diseño, Desarrollo, Instrumentación y Pruebas (Clark95) (Clark00).

1. Ordene Material: Epitomize chunk material.
2. Establezca secuencia lógica.
3. Construya un Artefacto interesante.
4. Organice los Objetivos.
5. Evocar Conocimientos Previos.
6. Actividades: fomente participación activa con preguntas, pida ejemplos..
7. Reflexionar y Sumarizar.
8. Retroalimentación y Evaluación.
9. Auxiliar transferencia de conocimientos.

Otros Modelos Prescriptivos Cognitivistas de Diseño: ^{xvi}

1. Algo-Heuristic Theory (Landa)
2. ASSURE Model
3. Criterion Referenced Instruction (Mager)
4. Morrison, Ross and Kemp Model

5. Dick and Carey Model
6. Elaboration Theory (Reigeluth, Merrill, Wilson & Spiller)
7. The 4C/ID Model (Jeroen van Merriënboer)
8. OEM - Organizational Elements Model (Roger Kaufman)
9. Rapid Prototyping (Tripp & Bichelmeyer)

4.3.2. Resumen de Modelos Instruccionales

Como se ha visto, existen diversos Modelos Instruccionales, entre los que se encuentran ADDIE, ASSURE, el modelo de Dick & Carey, el modelo de Gagné-Briggs, por mencionar algunos^{xvii}. Todos estos modelos tienen componentes comunes y están estructurados de manera muy similar a los métodos de ingeniería de software:

1. Identificar y analizar los objetivos instruccionales.
2. Planear y diseñar soluciones a los objetivos instruccionales.
3. Implementar una solución.
4. Evaluar y revisar los objetivos, estrategias, etc.

4.3.3. Estrategias de Diseño Instruccionales:

- ✓ **Organizacionales:** Planeación, estructuración y secuenciación lógica y didáctica de las actividades
- ✓ **Presentación (Delivery):** selección de medios, recursos y formato de presentación de los materiales
- ✓ **Gestión:** Seguimiento, asesoría, evaluación y registro de las actividades por cada participante.

4.4. De las teorías de aprendizaje

- c) El *Conductismo* se basa en los cambios observables en la conducta del sujeto. Se enfoca hacia la repetición de patrones de conducta hasta que se realizan de manera automática. La teoría del conductismo se concentra en el estudio de conductas que se pueden observar y medir (Good y Brophy, 1990). Ve a la mente como una “caja negra” en el sentido de que las respuestas a estímulos se pueden observar cuantitativamente ignorando totalmente la posibilidad de todo proceso que pueda darse en el interior de la mente. Algunas personas claves en el desarrollo de la teoría conductista incluyen a Pavlov, Watson, Thorndike y Skinner.
- d) El *Cognoscitivismo* se basa en los procesos que tienen lugar atrás de los cambios de conducta. Estos cambios son observados para usarse como indicadores para entender lo que está pasando en la mente del que aprende. “Los teóricos del cognoscitivismo reconocen que una buena cantidad de aprendizaje involucra las asociaciones que se establecen mediante la proximidad con otras personas y la repetición. También reconocen la importancia del reforzamiento, pero resaltan su papel como elemento retroalimentador para corrección de respuestas y sobre su función como un motivador. Sin embargo, inclusive aceptando tales conceptos conductistas, los teóricos del cognoscitivismo ven el proceso de aprendizaje como la adquisición o reorganización de las estructuras cognitivas a través de las cuales las personas procesan y almacenan la información.” (Good y Brophy, 1990, pp. 187).
- e) El *Constructivismo* se sustenta en la premisa de que cada persona construye su propia perspectiva del mundo que le rodea a través de sus propias experiencias y esquemas mentales desarrollados. Para Schuman el constructivismo se enfoca en la preparación del que aprende para resolver problemas en condiciones ambiguas. El pionero de la primera aproximación constructivista fue Barlett^{xviii} el *constructivismo* se sustenta en que “el que aprende construye su propia realidad o al menos la interpreta de acuerdo a

la percepción derivada de su propia experiencia, de tal manera que el conocimiento de la persona es una función de sus experiencias previas, estructuras mentales y las creencias que utiliza para interpretar objetos y eventos.” “Lo que alguien conoce es aterrizado sobre las experiencias físicas y sociales las cuales son comprendidas por su mente.” (Jonasson, 1991).

4.5. De la Ingeniería Web y el Diseño Instruccional

“La implementación de la tecnología significa que nuestros profesores, inicialmente, tendrían trabajo adicional. No sólo deberían aprender el uso de la nueva tecnología, sino también retribuir el currículum a partir de una forma totalmente diferente de enseñar.”

Resulta un tanto difícil compatibilizar un producto obtenido por desarrolladores de software y un software educativo. En general, los ingenieros Web o ingenieros de software, están preparados para desarrollar software, pero conocen poco de la parte educativa, muchos se rigen de la intuición ya que en general no reciben educación pedagógica.

Lo deseable es que el equipo de desarrollo conozca algunos conceptos que se han abordado en este capítulo. Para lograr una herramienta WebApp para la enseñanza se debe tener en cuenta: (1)el Diseño Instruccional para lograr un (2)Aprendizaje Significativo, teniendo en cuenta (3)las diferentes Teorías de Aprendizaje.

4.5.1. Beneficios que aporta el Diseño Instruccional a e-Learning [elearnspace].

- ✓ Tiende un puente entre el campo educativo y el tecnológico, manteniendo el énfasis en el aprendizaje en vez de la tecnología.
- ✓ Contribuye al éxito del aprendizaje del estudiante proporcionando una orientación más didáctica a las actividades y recursos involucrados.
- ✓ Permite asegurar la calidad, efectividad y consistencia de los materiales.
- ✓ Permite explotar apropiadamente el potencial de las distintas tecnologías educativas.
- ✓ Establece bases sólidas y formales para aplicar nuevas teorías y métodos Instruccionales.

4.5.2. Método de Especificación Instruccional de Software Educativo (EISE)

EISE es una propuesta, seguramente una de las más serias, del proyecto Galileo que intenta “industrializar” el software educativo.

De la necesidad de contemplar los aspectos educativos en el proceso de desarrollo de software, se desarrolló un método para especificar las necesidades, objetivos y metas de aprendizaje que los sistemas deberán cubrir.^{xix}

EISE se ha utilizado para especificar las nuevas herramientas de construcción y simuladores que se están desarrollando en el proyecto Galileo, lo cual ha permitido:

- ✓ Mejorar la comprensión del contexto educativo de las áreas de conocimiento.
- ✓ Mejorar el diseño conceptual y visual de los productos educativos.
- ✓ Mejorar la comunicación con los evaluadores institucionales a quienes se les han presentado los nuevos productos.
- ✓ Mejorar la comunicación con los profesores durante la capacitación y las pruebas de campo.

Para desarrollar un modelo propio, se adoptaron algunos de los principios del modelo de Dick & Carey, Gagné y ADDIE, además de tomar en cuenta algunos principios de diseño específicos a tecnología educativa, a manera que la especificación pueda plasmar:

- ✓ La estructura del área de conocimiento a explorar.
- ✓ La estructura de los planes de estudio del área de conocimiento del nivel académico correspondiente.

- ✓ Las actividades que estudiantes y profesores realizan.
- ✓ Los materiales instruccionales de soporte existentes y los que se tendrán que desarrollar.
- ✓ Los conocimientos previos del estudiante.
- ✓ El planteamiento de una metáfora (si aplica).
- ✓ La aplicación de patrones pedagógicos y estrategias de aprendizaje

4.5.2.1. La estructura del método EISE es la siguiente:

- ✓ **Análisis de contenidos:** Consiste en revisar los planes de estudio, contenidos, definiciones, tareas y herramientas usadas.
- ✓ **Análisis de necesidades:** Consiste en identificar las metas de aprendizaje del proyecto instruccional, así como las necesidades entre las metas y el estado actual.
- ✓ **Análisis instruccional:** Consiste en analizar las metas describiendo paso a paso como se lograrán, para determinar los conocimientos y habilidades que el usuario requerirá para realizar cada paso de la meta.
- ✓ **Diseño de objetivos y estrategias instruccionales:** Consiste en especificar los objetivos instruccionales, los eventos de aprendizaje (de acuerdo a Gagné) y las estrategias (o planes) para presentar la instrucción a los usuarios.
- ✓ **Diseño de la solución:** Consiste realizar una representación de la pantalla instruccional, especificar los principales detalles del modelo que hará funcionar al sistema, y listar los recursos adicionales necesarios (en esta etapa el DI se complementa con algunos principios de diseño de software educativo y técnicas de prototipado de baja fidelidad).
- ✓ **Evaluación:** Consiste en evaluar el aprendizaje y comportamiento del usuario, así como los efectos del uso de la solución.

Cabe resaltar que por su naturaleza, EISE toma en cuenta el trabajo de Diseño Instruccional hecho por diversas personas e instituciones en el área de conocimiento objetivo.

La siguiente figura muestra la estructura de EISE:

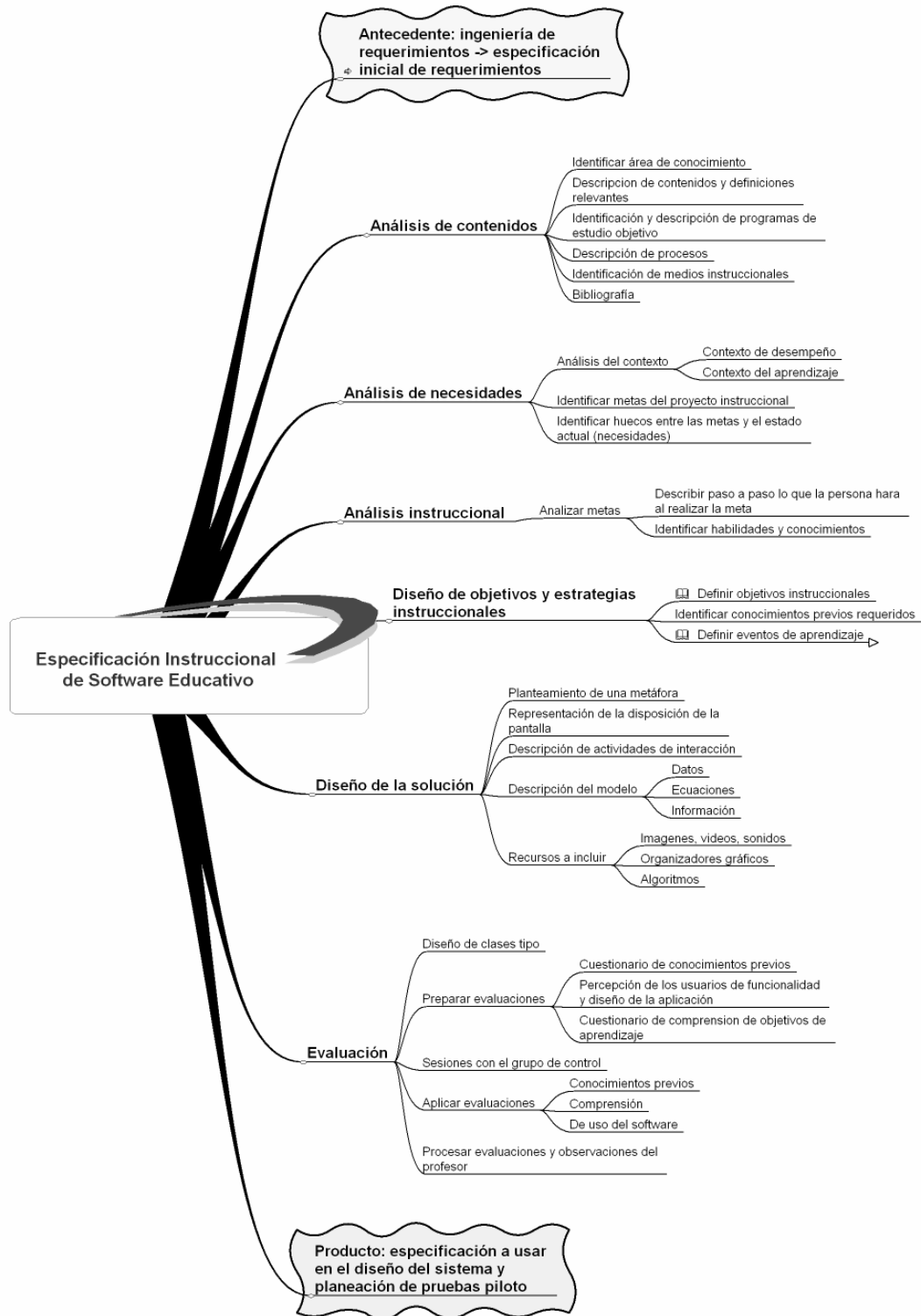


Figura 12 – Estructura de la especificación Instruccional de software educativo (Extraído de “Diseño Instruccional aplicado al desarrollo de software educativo – E. J. Valdemar)

Capítulo 5 – Conclusiones

Muchos profesores poseen materiales didácticos que utilizan para dictar sus asignaturas, es común preparar una clase en Power Point, y hasta brindan al estudiante una batería de ejemplos en soporte magnético. Sin embargo, no todos se atreven a construir cursos multimedia interactivos, en el que puedan adecuar e integrar el contenido de los materiales e implementar algunas estrategias de enseñanza-aprendizaje que promueva un aprendizaje significativo por parte de los estudiantes.

El docente, quizá desea usar herramientas como el moodle, wikis, blogs, pero se encuentra en una encrucijada: dedicar su tiempo a enriquecer la currícula o aprender un soft nuevo que le demandaría horas de trabajo (las primeras infructuosas), tomar un curso significa tiempo y dinero extra, siendo las primeras experiencias en el uso de estos software, bastante frustrante para quienes “recién comienzan”.

Compatibilizar Internet con el sistema educativo no es una tarea fácil. El proyecto Galileo 2003-2007 muestra la experiencia en el desarrollo de productos de software educativo como una industrialización de software “*el software educativo son programas de computadoras para facilitar los procesos de enseñanza y aprendizaje*”.

Lo deseable es que el equipo de desarrollo conozca y tenga en cuenta conceptos tales como:

1. El Diseño Instruccional
2. Aprendizaje Significativo
3. Las diferentes Teorías de Aprendizaje.

5.1. Estructura del Diseño Instruccional y de la Ingeniería de Software

Como se plantea en 4.3.2 el diseño instruccional se puede simplificar como la tarea de “planificación” de una currícula. Este plan contiene, al igual que la ingeniería de software, diferentes pasos que se pueden resumir en la siguiente estructura:

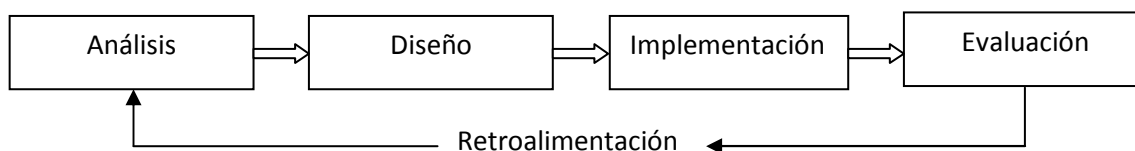


Figura 13: Etapas de la Ingeniería de Software coincidentes con las etapas del diseño instruccional.

DI Vs. IS	Análisis	Diseño	Implementación	Evaluación	Retroalimentación	
Diseño Instruccional	Identificar y analizar los Objetivos Instruccionales.	Diseñar soluciones.	Desarrollar Soluciones.	Evaluar los objetivos.	PROVEER RETROALIMENTACION	
	Modelo ADDIE					
	<ul style="list-style-type: none"> • Características de la audiencia • Qué necesita aprender • Presupuesto disponible • Medios de difusión • Limitaciones • Fecha límite para implantar la instrucción • Actividades para el logro de los componentes 	<ul style="list-style-type: none"> • Selección del ambiente • Señalamiento de los objetivos instruccionales • Selección de estrategias pedagógicas • Bosquejo de las unidades, lecciones y módulos • Diseño del contenido del curso (Ordenar Material, establecer secuencia lógica) 	<ul style="list-style-type: none"> • Construcción de artefactos interesantes. • Organizar Objetivos • Evocar conocimientos previos • Utilización de la red. • Determinar interacciones • Planificar actividades • Duplicación y distribución de materiales • Se implanta el curso • Se resuelven problemas técnicos y se discuten planes alternos. 	<ul style="list-style-type: none"> • Desarrollo de pruebas para medir estándares instruccionales. • Implantación de pruebas y evaluaciones. • Planificación de evaluaciones estudiantiles. • Desarrollo de evaluaciones formativas para evaluar el curso. • Evaluaciones sumativas para emitir juicio de la efectividad de la instrucción. 		
	Modelo de GAGNE					
<ul style="list-style-type: none"> • Cómo atraer atención (evaluar la población a la que va dirigido el material) • Establecer objetivos del aprendizaje • Analizar saberes previos (crear conocimiento propio) 	<ul style="list-style-type: none"> • Diseño de Materiales (que estimulen al aprendiz, teniendo en cuenta el paso anterior) • Guiar aprendizaje (ej: links a paginas de interés) • Establecer material adicional • Proveer retroalimentación (ej: foros, autoevaluaciones) 	<ul style="list-style-type: none"> • Experimentar o puesta en práctica. 	<ul style="list-style-type: none"> • Evaluación de transferencia o rendimiento 			
Ingeniería de Software	<ul style="list-style-type: none"> • Elicitación de Requerimientos • Análisis de Casos de Uso 	<ul style="list-style-type: none"> • Diseño Conceptual (establecer clases y relaciones) 	<ul style="list-style-type: none"> • Desarrollo de la aplicación en un lenguaje de programación determinado 	<ul style="list-style-type: none"> • Pruebas • Evaluación de resultados 		

Tabla 3: Tabla comparativa de las Etapas en el desarrollo de software y del Diseño Instruccional

5.2. Estructura del Diseño Instruccional y de la Ingeniería Web

La Ingeniería Web incorpora el diseño navegacional como una etapa crítica. El análisis de la navegación de los sitios web, permite replicarlos en una estructura de grafos como mapas conceptuales, estas estructuras son ampliamente utilizadas en modelos constructivistas y cognitivistas. Dicha coincidencia se puede apreciar claramente en el modelo de diseño de

hipermedia Orientado a Objetos (OOHDM) en sus diferentes etapas - apartado 2.3.4. de este documento.

5.2.1. Ejemplo de uso de la Ingeniería Web en el Diseño Instruccional

Una aplicación interesante y muy utilizada en la elaboración de materiales multimedia, es la generación de tutoriales.

El trabajo que se expone en 5.2.1.1, que fuera aprobado e invitado a ser expuesto en la “IV Jornada de innovación docente: Docencia y TICs” de la Universidad de Valladolid – España, se basa en los nueve eventos instruccionales de Gagné (condiciones internas) y en la definición de especificaciones dada por la corporación de e-learning: IMS Global Learning Consortium.

Dicho trabajo considera:

1. **Aspectos considerados importantes en el IMS Learning Design:**
 - ✓ Ofrecer soporte para múltiples alumnos, y contemplar la comunicación entre ellos
 - ✓ Representar el papel de profesor.
 - ✓ Permitir combinar recursos educativos con actividades pedagógicas, y con las interacciones entre personas en diferentes roles.
2. **Eventos Instruccionales:**
 - ✓ Atraer atención de los alumnos proporcionando elementos multimediales a manera de simulación de pasos para resolver un problema o cómo manejar un software.
 - ✓ Cada tema se presenta con un pequeño resumen (objetivo) y se divide en varias páginas, para no sobrecargar de información.
 - ✓ Evocar conocimientos dados en asignaturas de cursos anteriores, proporcionar vínculos a páginas de interés que ayude a generar su propio conocimiento.
 - ✓ Guiar el proceso de aprendizaje: proporcionando pistas cognitivas y enlaces a páginas de interés.
 - ✓ Provee retroalimentación: que se enriquece mediante el uso de foros y mails
 - ✓ Realiza autoevaluaciones.
 - ✓ Proporciona material adicional: ejercicios, ejemplos, etc.

5.2.1.1. Herramienta de apoyo a la docencia: tutorial interactivo personalizado

A fin de ayudar en el proceso de enseñanza-aprendizaje, se desarrolló un Tutorial con software libre que combina elementos curriculares teóricos, hipervínculos a páginas de interés, elementos multimediales, ejemplos prácticos y un foro que alienta la interactividad.

El uso de esta herramienta informática busca:

- a) *Reforzar los contenidos teóricos vertidos en clase.*
- b) *Guiar a los estudiantes en la búsqueda de información válida, que les sirva de base en la investigación y resolución de problemas por caminos alternativos.*
- c) *Que puedan compartir experiencias, dudas, aciertos y errores.*
- d) *Facilitar la tarea del docente: para que centre su atención en la elaboración de contenidos curriculares y no resulte una complicación adicional.^{xx}*

La aplicación, denominada ITE (Implementación para generar Tutoriales Educativos), contiene los siguientes módulos:

- ✓ **Asistencia:** Guía práctica de principios del diseño Instruccional de entornos de aprendizaje apoyados con TICs.
- ✓ **Base de datos:** de contenidos teóricos, elementos multimediales, ejemplos, autoevaluaciones, vínculos.
- ✓ **Administración:** brinda la posibilidad de agregar, modificar o borrar tutoriales y/o contenidos en uno creado.
- ✓ **Espacios de comunicación:**
 - Foro abierto: por tratarse de una universidad pública y gratuita, se prevé un foro abierto, moderado.

- **Foro de consulta:** cada tema contendrá un espacio para la consulta en la que podrán participar alumnos registrados.
- ✓ **Búsqueda:** posibilita encontrar información dentro de la base de datos.
- ✓ **Contacto:** ofrece una ventana de contacto directo alumno - docente.

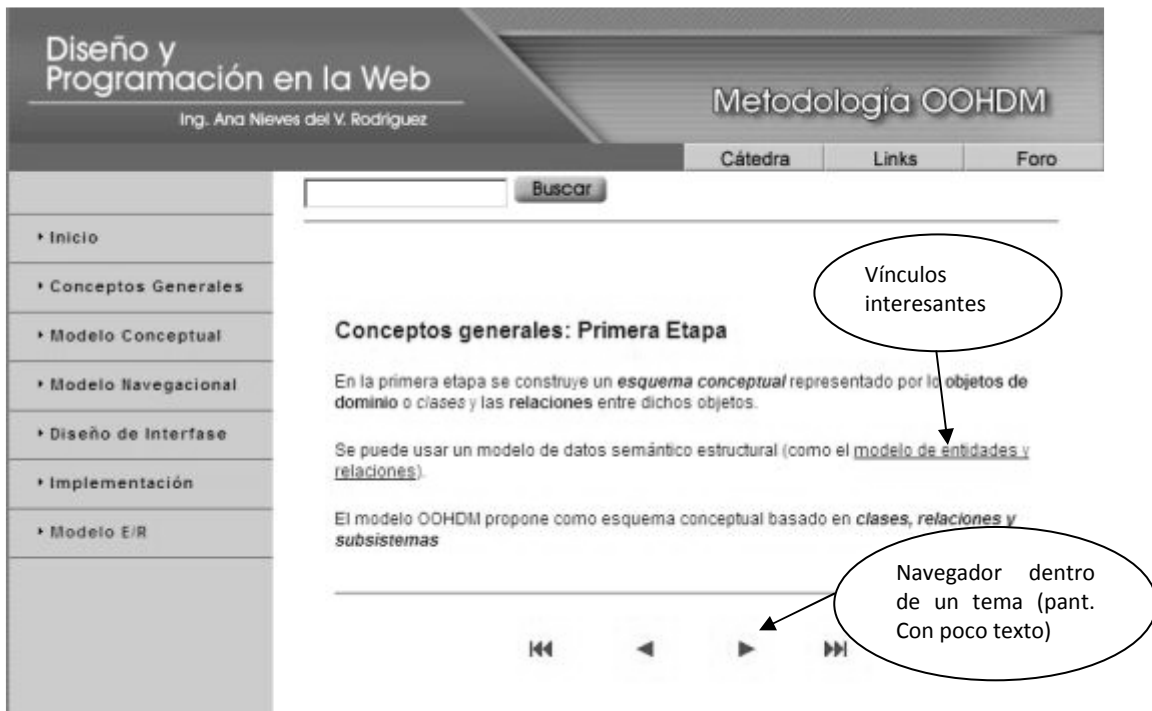


Figura 14: Pantalla de ITE para la asignatura Diseño en la Web

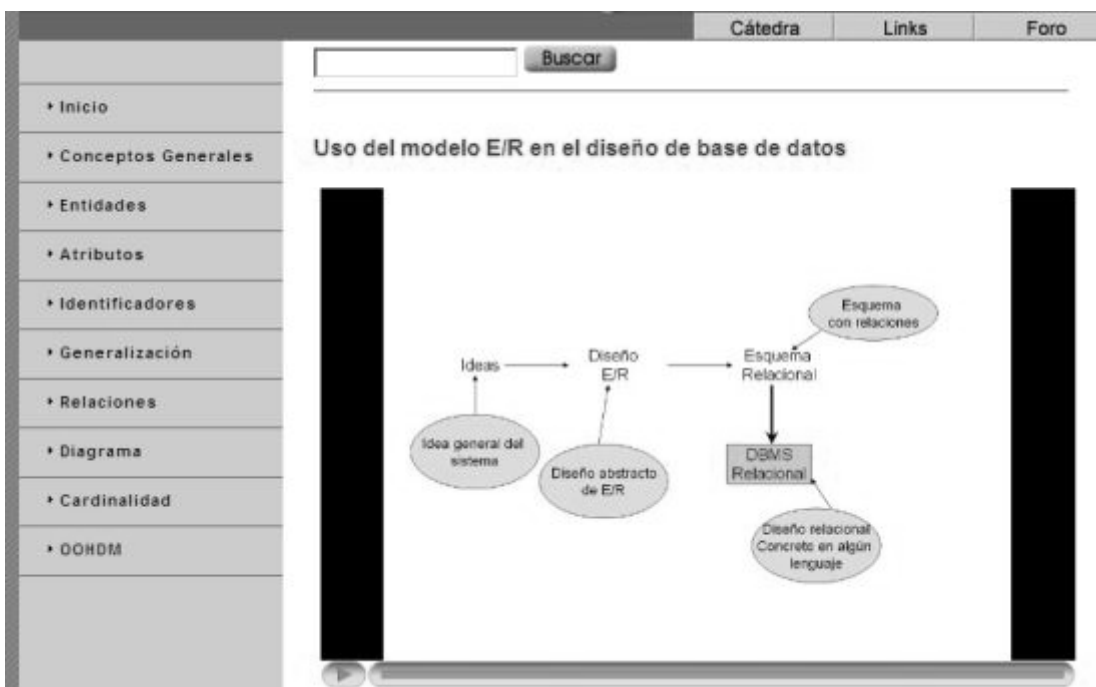


Figura 15: Elementos Multimediales

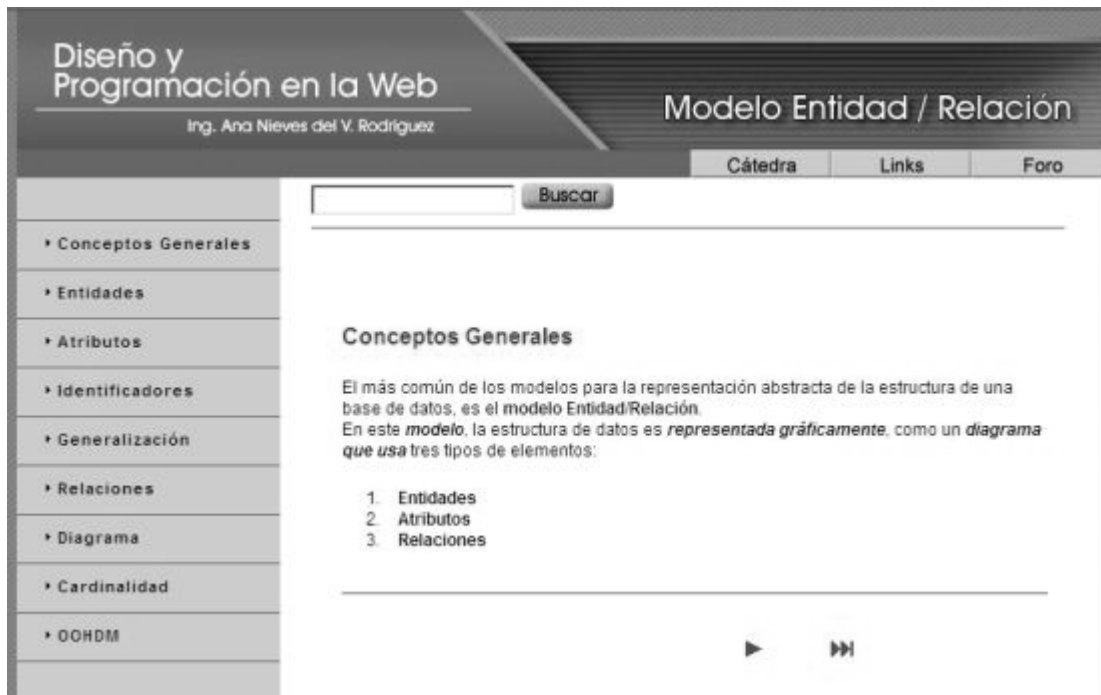


Figura 16: Pantalla que refiere a conocimientos previos

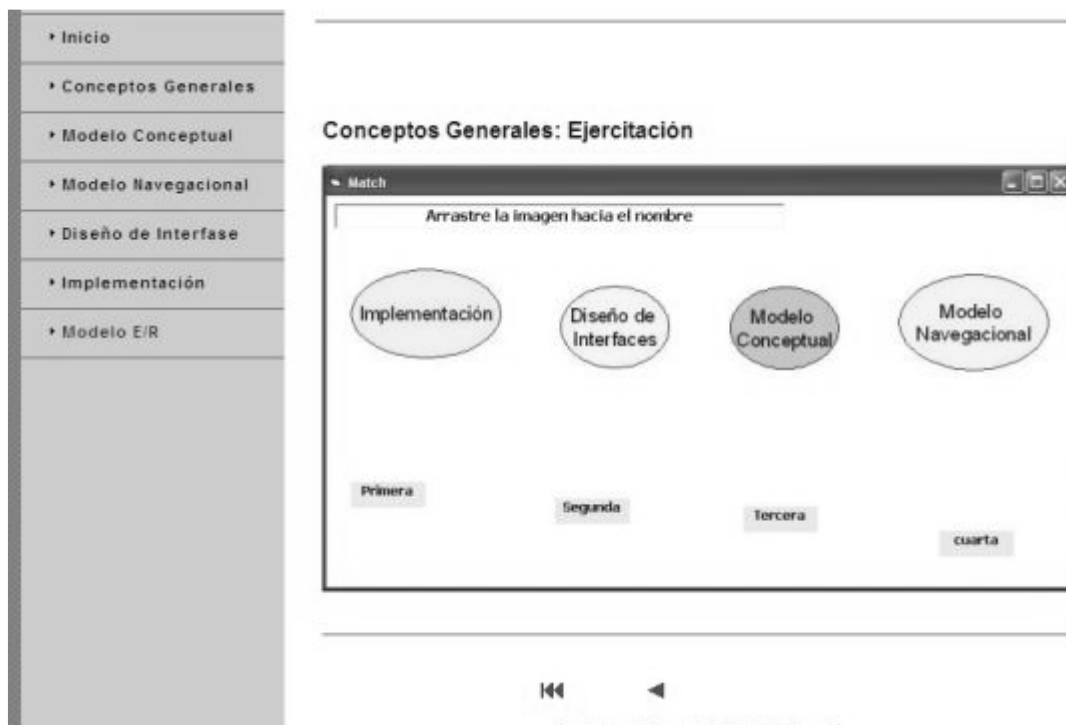


Figura 17: Pantalla de ejercitación

5.3. Consideraciones finales

En este trabajo, se abordó el tema del Diseño Instruccional su vinculación con la ingeniería Web, y como conclusión se muestra una aplicación que utiliza el punto de encuentro de ambos temas.

Por lo expuesto, la Ingeniería Web debería ser un aliado incondicional del proceso de enseñanza-aprendizaje, buscando lograr buenos productos: atractivos y seductores tanto para alumnos como para docentes.

Referencias y Bibliografía

- ⁱ La Ciencia Su método y su filosofía - Mario Bunge - 2007
- ⁱⁱ 1001 things everyone should know about science - Trefil James
- ⁱⁱⁱ Social change, identity shift and language shift of K'iche' in Guatemala. Ph.D. thesis. Georgetown University. xviii, 439 p – M. Paul Lewis - 1994
- ^{iv} "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13 – Cota A.
- ^v Software Craftmanship - Pete McBreen
- ^{vi} The Humble Programmer – Edsger Dijkstra
- ^{vii} "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, Vol. 20, No. 4 (April 1987) pp. 10-19 - Brooks, Frederick P.
- ^{viii} "La estructura de las revoluciones científicas" – Tomas Kuhn
- ^{ix} Ingeniería del software – Un enfoque práctico - Roger Pressman
- ^x Calidad en Productos Web – Luis Olsina - 30 JAIIO – 2001
- ^{xi} "RMM: A methodology for the design of structured hypermedia" - ISAKOWITZ, T.; STOHR, E.A.; BALASUBRAMANIAN, P. Communications of the ACM, vol. 38, 1995
- ^{xii} On Conceptual Modeling of Device-Independent Web Applications: Towards a Web Engineering Approach. ISSN:1070-986X. IEEE Multimedia 8(2). 2001. Page(s): 26-40 - Gómez J., Cachero C., Pastor O.
- ^{xiii} Web Site Engineering – Powell, T. A. - 1998
- ^{xiv} La generación multimedia - Roxana Morduchowicz
- ^{xv} El capital intelectual de los jóvenes - Roxana Morduchowicz
- ^{xvi} "The Many Problems of Representation" – Ryder, D. - 2008
- ^{xvii} Educational Multimedia: A Handbook for Teacher-Developers - Usha V. Reddi and Sanjaya Mishra, Editors. Commonwealth Educational Media Centre for Asia, March 2003. ISBN: 81-88770-00-0
- ^{xviii} Educational psychology: A realistic approach. (4th ed.).White Plains, NY: Longman - Good, T. L., Brophy, J. E. (1990).
- ^{xix} Diseño Instruccional Aplicado Al Desarrollo De Software Educativo - Eugenio Jacobo Hernández Valdelamar.
- ^{xx} "Innovación docente: docencia y TICs" – Universidad de Valladolid – ISBN: 978-84-691-5535-6 – Capítulo: "Herramienta de apoyo a la docencia tutorial interactivo personalizado", pp: 29-40; Ana Nieves Rodríguez, Pablo Gonzalo Costilla – Setiembre de 2008