

UNIVERSIDAD DE MONTEMORELOS
FACULTAD DE INGENIERÍA Y TECNOLOGÍA



PROPUESTA DE ESTÁNDAR DE SOFTWARE
DE EVALUACIÓN DEL APRENDIZAJE
PARA LA EDUCACIÓN VIRTUAL

TESIS

PRESENTADA EN CUMPLIMIENTO PARCIAL DE LOS
REQUISITOS PARA EL GRADO DE MAESTRÍA EN
CIENCIAS COMPUTACIONALES

POR

ALBERTO MONCADA GIL

CIB
Ej.1



65323

AGOSTO DE 2004

UNIVERSIDAD DE MONTEMORELOS
FACULTAD DE INGENIERÍA Y TECNOLOGÍA



**PROPUESTA DE ESTÁNDAR DE SOFTWARE
DE EVALUACION DEL APRENDIZAJE
PARA LA EDUCACIÓN VIRTUAL**

TESIS

**PRESENTADA EN CUMPLIMIENTO PARCIAL DE LOS
REQUISITOS PARA EL GRADO DE MAESTRIA EN
CIENCIAS COMPUTACIONALES**

POR

ALBERTO MONCADA GIL

AGOSTO DE 2004

065323



RESUMEN

PROPUESTA DE ESTÁNDAR DE SOFTWARE DE EVALUACIÓN DEL APRENDIZAJE PARA LA EDUCACIÓN VIRTUAL

por

Alberto Moncada Gil

Asesor principal: Saulo Hernández Osoria

RESUMEN DE TESIS DE POSGRADO

Universidad de Montemorelos

Facultad de Ingeniería y Tecnología

TÍTULO: PROPUESTA DE ESTÁNDAR DE SOFTWARE DE EVALUACIÓN DEL APRENDIZAJE PARA LA EDUCACIÓN VIRTUAL

Investigador: Alberto Moncada Gil

Asesor principal: Saulo Hernández Osoria, maestro en Ciencias en Tecnología Informática.

Asesores: Andrés Díaz Valladares, maestro en Educación y maestro en Tecnología de la Información; y Ana Lucrecia Salazar Rodríguez, doctora en Educación.

Fecha de terminación: Agosto de 2004

Problema

La presente investigación propone un estándar de la ingeniería del software para el software educativo diseñado para crear, aplicar y ponderar actividades de evaluación del aprendizaje en la educación virtual.

Método

La metodología fue esencialmente documental. Se procedió a revisar en la literatura los conceptos fundamentales del dominio del problema, la pedagogía, y del dominio de la solución, la ingeniería del software. A partir de ellos se elaboró un modelo pedagógico de

evaluación aplicable a la educación virtual, vía internet, y se definió un proceso para producir un estándar informático.

Del modelo pedagógico de evaluación se obtendrían los requerimientos de software para un sistema hipotético de evaluación del aprendizaje en línea. Entonces los requerimientos servirían de base para un proceso de especificación de cuatro etapas iterativas: (a) análisis de requerimientos o definición del problema; (b) diseño o definición de una solución (bosquejo de la propuesta); (c) revisión de la propuesta; y (d) Formulación del estándar.

Siguiendo la tendencia en cuanto a este tipo de especificaciones, el producto final, la propuesta de estándar, se expresó como un conjunto de esquemas de XML o XSD.

Resultados

Siguiendo la metodología descrita, se obtuvieron los siguientes productos:

1. El esquema de un modelo pedagógico desde el punto de vista de la evaluación del aprendizaje.
2. Un proceso para la creación de estándares.
3. Una propuesta de estándar para software de evaluación del aprendizaje expresado en dos esquemas de XML, relacionados con otros seis esquemas de XML auxiliares, que simulan las propiedades de todo un sistema de aprendizaje o *learning management system* (LMS). Los dos esquemas de XML corresponderían a un subsistema de cualquier LMS.

Tal como lo indica su uso, estos esquemas de XML pueden aplicarse tanto para definir la estructura de datos de un sistema, como para servir de plantilla o validar conjuntos de datos que se transmitan entre sistemas que interoperen.

Adicionalmente, se ha expresado la propuesta de estándar en forma de lista de revisión como una guía para evaluar de manera general software de este tipo.

Conclusiones

La propuesta resultante constituye una guía para el desarrollo de software de evaluación del aprendizaje, ya que ofrece los requerimientos generales y la estructura de datos que los satisfacen, aunque deja en libertad a los desarrolladores para decidir la mejor implementación. Puede también servir de guía para la revisión o evaluación de software ya existente, que pretende actualizarse apuntando a la interoperabilidad. Y, finalmente, los esquemas de XML pueden utilizarse para crear capas de interoperación en sistemas heterogéneos que requieran comunicarse entre ellos, sin necesidad de grandes adaptaciones.

Universidad de Morelos

Facultad de Ingeniería y Tecnología

PROPUESTA DE ESTÁNDAR DE SOFTWARE
DE EVALUACIÓN DEL APRENDIZAJE
PARA LA EDUCACIÓN VIRTUAL

Tesis
presentada en cumplimiento parcial de los
requisitos para el grado de Maestría en
Ciencias Computacionales

por

Alberto Moncada Gil

Agosto de 2004

**PROPUESTA ESTÁNDAR DE SOFTWARE DE EVALUACIÓN DEL
APRENDIZAJE PARA LA EDUCACIÓN VIRTUAL**

**Tesis
presentada en cumplimiento parcial
de los requisitos para el grado de
Maestría en Ciencias Computacionales**

**Por
Alberto Moncada Gil**

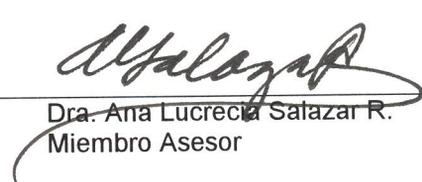
APROBADA POR LA COMISIÓN:


Mtro. Saulo Hernández Osoria
Asesor Presidente


Mtro. Jorge Manrique
Director Facultad Ingeniería y
Tecnología


Mtro. Andrés Díaz Valladares
Miembro Asesor


Dra. Raquel Korniejczuk
Directora de estudios graduados


Dra. Ana Lucrecia Salazar R.
Miembro Asesor

02 DE AGOSTO DE 2004

Fecha de Aprobación

DEDICATORIA

Al Señor, quien puso en mí sed de conocimiento.

A mi esposa, que cree en mí.

TABLA DE CONTENIDO

LISTA DE TABLAS Y FIGURAS	vii
PREFACIO	viii
Capítulo	
I. INTRODUCCIÓN	1
Antecedentes	1
Los estándares del software	1
Utilidad de los estándares	2
Conjunción de estándares de software y educativos	3
La evaluación	5
Software educativo para evaluación	6
Cómo crear un estándar	7
Declaración del problema	8
Justificación	8
Objetivos	9
Delimitaciones	9
Preguntas de investigación	10
Definición de términos	10
II. MARCO TEÓRICO	13
El modelo pedagógico	13
El enfoque pedagógico moderno	14
Constructivismo	16
La educación virtual	20
Introducción de la computadora	20
Adaptación al nuevo entorno	23
Modelos para la educación virtual	26
Puntos álgidos del elearning	31
Evaluación del aprendizaje	33
Crítica a los exámenes tradicionales	34
Clases de evaluación	36
Evaluación alternativa	37
Métodos de cálculo de la calificación	39
Autoevaluación y realimentación	43
Evaluación en línea	44

Estándares de software	46
Definiciones y aportaciones	46
Definiciones	48
Cualidades de un buen estándar	48
Necesidades satisfechas por los estándares	49
En busca de calidad	51
Historia y organizaciones	52
Principales organizaciones estandarizadoras	54
Estandarizadores de la educación virtual	58
Estándares del elearning	61
Establecimiento de estándares	67
El proceso de la JTC1 de la ISO/IEC	69
La codificación	71
La certificación	78
Estado actual de la estandarización en la ingeniería del software	80
Tendencias y proyectos	83
III. METODOLOGÍA	86
Proceso de desarrollo de un estándar	87
Análisis de requerimientos o definición del problema	87
Diseño y definición de una solución: Propuesta de especificación	88
Revisión de la propuesta	89
Formulación del estándar	89
Entregables	90
Documentación	90
Diagramas UML	90
Esquemas XML	91
Lista de revisión o checklist	91
IV. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS	92
Presuposiciones para describir e implementar los requerimientos	93
Análisis de requerimientos o definición del problema	93
Obtención de requerimientos: Modelo lógico	97
Requerimientos generales de un sistema de educación virtual	98
Requerimientos de un subsistema de evaluación del aprendizaje en la educación virtual	100
Diseño arquitectónico o definición de la solución: El Estándar	102
Entregables	108
Lista de revisión	109
V. CONCLUSIONES	112
Cumplimiento de objetivos	113
Implementación del estándar	115

Paso 1. El LMS	115
Paso 2. Estructura de datos del evaluador	115
Perspectivas de desarrollo	118
Apéndice	
A. CÓDIGO FUENTE DE LOS ESQUEMAS XML	119
B. DOCUMENTACIÓN DE LOS ESQUEMAS XML DEL SISTEMA EVALUADOR Y ESQUEMAS ACCESORIOS	126
C. EJEMPLOS DE APLICACIÓN DEL ESTÁNDAR	203
D. RECURSOS WEB Y CD ANEXO	209
REFERENCIAS	211

LISTA DE TABLAS Y FIGURAS

Tablas

1. Clases de evaluación	36
2. Comparación entre la evaluación alternativa y la evaluación tradicional	38
3. Métodos de cálculo de la calificación	40
4. Ejemplo de rúbrica	42
5. Habilidades de los sistemas de elearning que los estándares ayudan a asegurar	52
6. Principales organismos productores o certificadores de estándares relacionados con la ingeniería del software	53
7. Fases típicas del proceso de producción iterativa de estándares	68
8. Fragmento de las especificaciones del SIF	74

Figuras

1. Modelo de misión infinita creado para guiar el diseño y desarrollo del currículo y la instrucción en línea	28
2. Organización estructural de los estándares de la IEEE	72
3. Proceso para utilizar el XML	78
4. Esquema del modelo pedagógico de evaluación del aprendizaje	97
5. Casos de uso del estándar	103
6. Diagrama de clases de UML de los objetos que componen un LMS	104
7. Diagrama de clases con atributos de clase	106
8. Interoperabilidad entre sistemas heterogéneos mediante el estándar	114

PREFACIO

El interés por el tema de esta investigación surgió al trabajar en el desarrollo del LMS de la Universidad de Morelos, el e42 (<http://e42.um.edu.mx>). El gusto por aplicar exámenes sui géneris en mi experiencia docente y que me correspondiera el desarrollo del módulo de exámenes en las versión 2 y 3 del e42 me exigió un respaldo más sólido en el terreno pedagógico y la definición más puntual de las propiedades que las evaluaciones del aprendizaje deben poseer en la educación virtual. Así fue que acabé comprometiéndome de lleno con el *elearning*. No siempre es recomendable que el ingeniero de software se comprometa tanto con el dominio de un problema que debe resolver, pero en el caso presente resultó más que justificado.

Dios hizo posible el tiempo, los recursos y las circunstancias para que esta investigación fuera posible; puso a mi alcance las enseñanzas del maestro Saulo Hernández y del maestro Andrés Díaz, así como las didácticas y sabrosas charlas que sostuve con la doctora Ana Lucrecia Salazar. Mi jefe inmediato, el rector de la universidad, el profesor Ismael Castillo contribuyó enormemente al darme las facilidades laborales para dedicar tiempo a estos estudios.

Pero nada hubiera sido posible sin la maravillosa ayuda, comprensión y apoyo de mi esposa, la doctora Lourdes Lozano, y sin la prolongada paciencia de nuestros hijos, Claudia Mitzi y Edrei Alberto. Mi gratitud y amor para ellos.

CAPÍTULO I

INTRODUCCIÓN

Antecedentes

Los estándares del software

La formulación y aplicación de estándares es antigua; algunos de los organismos actualmente responsables de la estandarización son centenarios o han heredado la tradición de instituciones del siglo XIX o principios del siglo XX, según lo describe Moore (2000). La fuerte corriente de estándares existentes facilitó la creación de los primeros estándares en el área de la ingeniería del software, no obstante que la informática es una ciencia relativamente nueva.

Como ha ocurrido con frecuencia, estos inicios se dieron en el campo militar, siendo el Departamento de Defensa de Estados Unidos el creador del primer estándar de software en 1974. Seguirían luego reconocidas instituciones internacionales como la International Organization for Standardization (ISO) y la Electronic Industries Association (EIA), o en Estados Unidos el Institute of Electrical and Electronics Engineers (IEEE), que produjo su primer estándar de esta área en 1979 (Moore, 2000).

Precisamente en el ámbito de las tecnologías de la información (TI) la ISO y la International Electrotechnical Commission (IEC) trabajaban sobre terrenos comunes; para optimizar sus esfuerzos ambos organismos crearon en 1987 un grupo de trabajo conjunto, el Joint Technical Committee (JTC1), cuyo propósito es tratar con los requerimientos especiales

de las TI, con la tarea de establecer la estandarización necesaria. De las 19 subcomisiones del JTC1, el séptimo, o SC7, es el responsable de los estándares concernientes a la ingeniería del software (Rehesaar, 1996).

Con todo, los estándares no han recorrido un camino uniforme, ni siempre exitoso; una de las principales razones es la propia naturaleza de la ingeniería del software y el hecho que carezca de un cuerpo de conocimiento plenamente reconocido. Eso ha producido cientos de estándares de decenas de organizaciones de todo tipo (Moore, 2000).

Se admite que un estándar no puede desarrollarse en el caos, y que si se busca la estandarización tendría que eliminarse dicho caos primero (Abran, 1996, p. 140). Algunos ven todavía cierto desorden en la ingeniería del software, ya que no ha madurado lo suficiente. Esto explicaría las inconsistencias en este campo. “Mientras los estándares son una parte aceptada de las disciplinas maduras, tales como la ingeniería (física) y muchas de sus ramas, aún no se han establecido del todo en el área de la ingeniería y desarrollo de software” (Rehesaar, 1996, p. 123).

Utilidad de los estándares

A pesar de todo, se consideran necesarios los estándares. Sin ellos, por ejemplo, la internet se convertiría en una Torre de Babel, porque son como el fundamento del lenguaje común sobre el que construimos la diversidad del mundo (Rosen, 2002, p. 16). La estandarización es la herramienta independiente más poderosa para alcanzar calidad en un ambiente de información (Smith, 1997, p. 283). Añade Moore (2000) que “los estándares de la ingeniería del software pueden desempeñar una variedad importante de papeles para una organización” (p. 14), como convenir en nombres sucintos para concepto complicados,

describir las mejores prácticas, certificar dichas prácticas y sus productos y facilitar los acuerdos contractuales.

Lutz (2002) ilustra las ventajas del uso de estándares, cuando habla de la documentación de un proyecto:

Una de las motivaciones para especificar los requerimientos formalmente de cierta notación [estándar] es que hace más fácil y exacta la revisión, diseño, implementación y desarrollo de casos de prueba. La documentación formal también ha mostrado que mejora la calidad del producto final. (p. 216)

Conjunción de estándares de software y educativos

La estandarización tampoco es nueva en el campo de la evaluación educativa. En Estados Unidos, en 1987, se comenzaron a desarrollar un conjunto de estándares para certificar la competencia de los docentes en la evaluación de sus alumnos. “Los estándares están basados en dos supuestos: (a) evaluar al estudiante es una parte integral del papel del maestro, y (b) la buena enseñanza y la buena evaluación van juntos” (Cunningham, 1998, Introducción).

Sería natural esperar que los estándares del software y los estándares educativos convergieran dado que ambos campos tienen una larga historia de interacción que se remonta a la segunda generación de computadoras, a fines de los 50. “Alrededor de ese tiempo, las principales universidades comenzaron a usar computadoras con propósitos administrativos como la contabilidad, la nómina y el registro estudiantil. Al mismo tiempo la gente comenzó a usar computadoras para investigación instruccional” (Alessi y Trollip, 1985, p. 47).

Sin embargo, añaden Alessi y Trollip que no fue sino hasta la introducción de las microcomputadoras, en 1977, que el panorama de sistemas instruccionales por computadoras rebasó el ámbito de las grandes universidades y corporaciones.

Pero estos 25 años de asociación no han sido suficientes para madurar los estándares esperados, si bien existen algunos estándares de la ingeniería del software aplicables a la educación. Una de las organizaciones más renombradas en este terreno es el Instructional Management Systems Global Learning Consortium (IMS), otro más es el Schools Interoperability Framework (SIF), que en sus propias palabras es “el lugar donde compañías y educadores pueden participar del desarrollo de especificaciones que aseguren la interoperación en los ambientes instruccional y administrativos del nivel elemental” (Software and Information Industry Association, 2003).

Tanto el IMS como el SIF pretenden estándares para la interoperabilidad, es decir, para la comunicación de sistemas educativos heterogéneos. Pero en la práctica el desarrollo de un estándar para el software educativo parece ser más una carrera comercial, cuyo ganador será un estándar *de facto* por su dominio del mercado. Blackboard es uno de los competidores y afirma su pretensión de convertirse en una suerte de sistema operativo de corte instruccional (Blackboard, 2000).

No es del todo extraño que esto ocurra con el software educativo. Para la educación en línea no hay un modelo pedagógico universalmente aceptado; aún hay preguntas acerca de la integración de los dispositivos electrónicos en el aula, el impacto de las evaluaciones en línea y cómo aprovechar en general la tecnología web en el aula. Siguen en el aire las preguntas de si los cursos tradicionales deben rehacerse o hay una fórmula para adaptarlos al medio electrónico (véase Campbell y Pargas, 2003; Rupf, 2002; Woit y Mason, 2003).

A pesar de lo mucho que debe definirse, parece ir quedando en claro un propósito general: que hay diferencias entre la educación tradicional y la educación en línea; estas diferencias se admiten a grandes rasgos: “El cambio de paradigma es ir de un ambiente de

enseñanza a un ambiente de aprendizaje” (Berge y Collins, 1995, p. 4). Aunque el modo de implementar este nuevo paradigma dista aún de ser totalmente maduro, en parte porque el mismo vehículo informático lo es.

La evaluación

En cierto sentido resulta una ventaja para la enunciación de estándares educativos, que al menos la estructura básica de una actividad de evaluación esté bien definida. Lo que está a discusión es su implementación, cómo ponderarlo, el significado de sus resultados y su peso en la evaluación final de un estudiante. “De hecho, el examen es objeto de continuas polémicas. En estricto sentido no pertenece a ninguna disciplina; evidentemente tampoco pertenece a la pedagogía” (Díaz, 1993, p. 7).

La incorporación de sistemas informáticos a la evaluación resalta esta problemática. Baste por ejemplo enunciar que se aceptan dos formas básicas de evaluación: formativa y sumativa, siendo el propósito de la primera “determinar el grado de dominio de una tarea de enseñanza dada y descubrir la parte de la tarea que no se domina”, en cambio, la evaluación sumativa se dirige a la obtención de una calificación general del aprovechamiento (Bloom, Madaus y Hastings, 1981, p. 71). La pregunta es: ¿Cómo distingue el software entre ambos? Esta cuestión sigue estando del lado del educador; un sistema puede calificar automáticamente los exámenes objetivos, pero debe dejar al instructor los ítems de construcción, como las preguntas abiertas o de desarrollo (Cunningham, 1998).

Debe admitirse, sin embargo, que el software también ha puesto al alcance de la educación algunas soluciones importantes. Cunningham afirma que

muchos maestros carecen de las destrezas necesarias para usar efectivamente las técnicas de evaluación... Los principios de medición son técnicos, matemáticos y oscuros. Aquellos que no pertenecen a este campo frecuentemente encuentran difícil

descifrar el código. Incluso cuando los procedimientos de evaluación son bien entendidos, la implementación puede consumir mucho tiempo (Introducción).

Ahora, con el uso de sistemas computacionales, bastaría con que el maestro conozca los principios de la evaluación para elegir una fórmula y dejar al software el aplicarla. Tal sería el caso de esquemas de evaluación complejos, como los que pretenden calificar al estudiante en función del rendimiento del grupo escolar, usando, por ejemplo puntuaciones z, rangos percentiles o stanines (Lafourcade, 1969).

Software educativo para evaluación

La delimitación de los elementos objetivos de las actividades de evaluación ha permitido la creación de gran cantidad de software con esa función, aunque no exista un estándar enfocado particularmente a ese tipo de sistemas.

En lo que concierne a la evaluación el software educativo existen dos tipos: (1) integrales o creadores de cursos, aquellos que ofrecen todo el conjunto de herramientas para crear, administrar, impartir y evaluar cursos en línea; y (2) los evaluadores, que son aplicaciones dirigidas casi exclusivamente a crear, aplicar y calificar exámenes. Los protagonistas del primer tipo son Blackboard y WebCT, los cuales controlan actualmente el 85% del mercado de la educación superior¹.

Con el fin satisfacer las necesidades de la educación en línea, son más convenientes los sistemas integrales, ya que “la evaluación ha de ser integral (del proceso de enseñanza – aprendizaje en su totalidad y de sus resultados; de lo cognitivo y lo afectivo; del diseño tanto curricular como informático, su ejecución y evaluación) y activa en tanto se trabaje la auto y coevaluación” (Gómez, Herrera y Bermúdez, 2001). Desarticular una actividad de evaluación

¹http://socrates.berkeley.edu/~fmb/articles/web_based_lms.html, recuperado el 2 de diciembre de 2003.

de un contenido podría tener cierto sentido sólo en cursos tutoriales o con fines clasificatorios o diagnósticos muy generales.

Cómo crear un estándar

Para estandarizar las aplicaciones educativas puede aplicarse el marco general en el que se desarrollan los estándares de la ingeniería de software. Sin ser parte de un proceso generalizado, tres elementos han ido afirmándose como protagonistas en el ámbito de la estandarización:

1. La metodología que aplican los organismos más importantes, como el IEEE y la ISO.
2. El uso de XML para la documentación, como lo hace la IMS. XML (eXtensible Markup Language) es un subconjunto del Standard Generalize Markup Language (SGML), un estándar de la ISO para definir y usar formatos de documentos. Mediante su uso puede capturarse tanto el contenido como el significado del documento (Hartmann, Huang y Tilley, 2001). Como un importante coadyuvante puede mencionarse también la posibilidad de modelar el estándar con UML (Medvidovic, Rosenblum, Redmiles y Robbins, 2002).
3. La expectativa general respecto a un estándar de ingeniería del software. Se esperaría que los estándares de la ingeniería de software contribuyan a la implementación de un modelo para el desarrollo, operación y administración de los sistemas de software (Moore, 2000). Así como se concluye que un producto que ha cumplido con ciertos estándares debe mostrar calidad, se presupone que los estándares son de calidad; ¿y qué quiere decir esto? Que poseen calidad técnica, pueden implementarse de manera práctica y eficiente, ofrecen soluciones y son oportunos, por mencionar sólo algunas cualidades (Updegrove, 1995).

Este documento toma en cuenta estos tres elementos para formular un estándar de la ingeniería del software para aplicaciones del campo educativo.

Declaración del problema

La presente investigación propone un estándar de la ingeniería del software para el software educativo diseñado para crear, aplicar y ponderar actividades de evaluación del aprendizaje en la educación virtual.

Justificación

Independientemente de su alcance y complejidad, el software educativo se relaciona siempre con actividades de evaluación, ya sea indirectamente al proponer el ambiente y los contenidos acerca de los cuales se quiere aplicar evaluaciones, o directamente al ser su función primaria ofrecer herramientas para crear, aplicar y ponderar tareas y exámenes. Esta conjunción informática y pedagógica se ha ido nutriendo de un desarrollo importante en las más diversas áreas de la tecnología de la información, particularmente la relacionada con los estándares de la ingeniería del software y las herramientas de desarrollo para el ambiente de internet. Puede verse que mientras se afina el fundamento conceptual de la educación a distancia por medio de internet, también se mejoran las herramientas de desarrollo, los lenguajes y el hardware.

Ya que hay una imparable y decidida corriente pedagógica inundando la internet, es apropiado ofrecer, desde el punto de vista de la ingeniería del software, criterios para crear o valorar el software educativo. Y puesto que hoy existen alternativas novedosas muy atractivas para producir un software superior, es aún más necesario que ciertos criterios estables normen el desarrollo, de modo que aun cuando se creen nuevos productos o se mejoren los existentes

incorporando las novedades tecnológicas, la base conceptual y la estructura lógica del software educativo permanezcan constantes. La estabilidad buscada puede descansar en un estándar o conjunto de estándares bien formulados, oportunos y basados en las necesidades reales de la ingeniería del software y la educación virtual.

Objetivos

Para el presente estudio se plantean los siguientes objetivos:

1. Presentar los elementos básicos del modelo pedagógico de evaluación del aprendizaje para la educación virtual.
2. Exponer el proceso para crear un estándar informático en la actualidad.
3. Formular un estándar para el software de evaluación del aprendizaje, satisfaciendo el modelo pedagógico descrito y el proceso de creación de estándares, con el cual sea posible:
 - a. Certificar la calidad del software educativo para evaluación del aprendizaje
 - b. Evaluar la oferta de software educativo
 - c. Dirigir el proceso de creación de nuevo software y fundamentar las pruebas a las que debe ser sometido
 - d. Facilitar la comunicación entre los agentes que coinciden alrededor del software educativo de evaluación: ingenieros, productores, pedagogos y clientes
 - e. La interoperabilidad entre aplicaciones de evaluación del aprendizaje

Delimitaciones

Con el fin de alcanzar una solución oportuna y práctica al problema planteado, la investigación se delimitó de la siguiente manera:

Se adoptó como adecuado un modelo pedagógico moderno que se aplica a la educación virtual, sin entrar en argumentaciones pedagógicas.

Aunque los exámenes tienen sus particularidades, estos y todo tipo de actividades de evaluación se consideraron informáticamente equivalentes. De modo que, para los propósitos del estándar propuesto, los requerimientos de un examen se aplicaron a los de cualquier actividad de evaluación.

Preguntas de investigación

Las siguientes preguntas dirigieron esta investigación:

P1. ¿Cuáles son las características del modelo pedagógico para crear e impartir cursos en línea?

P2. ¿Cuáles son las características de la evaluación del aprendizaje que se desprende del modelo pedagógico definido?

P3. ¿Cuáles son las características distintivas de un estándar moderno?

P4. ¿Qué procesos se siguen para proponer y establecer estándares de la ingeniería del software?

P5. ¿Qué requerimientos debe cumplir el estándar para software de evaluación del aprendizaje que satisfaga el modelo pedagógico de evaluación del aprendizaje?

P6. ¿Qué estándar se puede proponer para el software de evaluación del aprendizaje?

Definición de términos

Estándar: “Modelo, tipo, patrón, nivel” (*Diccionario de la Real Academia de la Lengua Española*, s. v. “estándar”). Conjunto de especificaciones técnicas que debe cumplir un programa, aplicación o software para certificarlo o considerar que ha cumplido con el

estándar. El estándar especifica cómo modelar los datos, la forma en que entran y se guardan los datos, medidas de validación de datos y de acceso a ellos, cómo debe transmitirse y desplegarse los datos, etc.

Un estándar puede ser (1) un objeto o medida de comparación que define o representa la magnitud de la unidad; (2) una caracterización que establece tolerancias o requerimientos permitidos por categorías de ítems; o (3) un grado o nivel de excelencia o cumplimiento requerido. Los estándares son definitorios por naturaleza, establecidos ya sea para tener una mayor comprensión e interacción o para reconocer características conductas observadas o normas deseables (Moore, 2000, p. 6).

Software educativo. Sistemas informáticos o programas de cómputo que gestionan actividades de enseñanza y aprendizaje o todo un ambiente de educación. Los sistemas complejos, integrales, se conocen genéricamente como *Learning Management Systems* o LMS.

Evaluación del aprendizaje. Conjunto de eventos y observaciones tendientes a confirmar que se ha aprendido, mediante la verificación del nivel de rendimiento de un alumno respecto a un dominio bien definido de conductas identificadas de un modo simple y de conveniencia con el conocimiento (Glabán y Ortega, 2002; Álvarez, 2001; Lafourcade, 1969). No es igual a medir y es más que sólo calificar; es interpretar una medida en relación con una norma preestablecida (Cunningham, 1998; Wiggins, 1998; Pérez, 1997).

Educación en línea. Conocida también como educación virtual, elearning y teleformación, es el proceso educativo que utiliza la internet como medio principal; puede ser tanto *online*, en línea, conectado activamente a internet, como *offline*, desconectado de

internet. Además, puede incluir el auxilio de otros medios electrónicos, como aplicaciones de escritorio, CD, etc.

Internet. Red de cómputo de extensión mundial.

CAPÍTULO II

MARCO TEÓRICO

El modelo pedagógico

“Lo pedagógico debe tener prioridad en relación con lo tecnológico” (Gómez, Herrera y Bermúdez, 2001). En efecto, para que la ingeniería del software, como ciencia instrumental al servicio de la educación, sea capaz de ofrecer soluciones eficaces, útiles y de calidad, debe comenzar por fundamentarse en un modelo pedagógico. Hablando en particular de educación virtual, algunos creen que internet por sí misma es el factor clave para el éxito, sin embargo, investigaciones al respecto proveen evidencia para afirmar que no es el medio el determinante, sino el diseño pedagógico usado en conjunción con las herramientas del medio (Koper, 2001).

¿Qué es este modelo para el diseño pedagógico, por qué tiene importancia y por qué debería preceder a la propuesta informática?

El modelo pedagógico no es un ideal, ni un camino predeterminado que hay que seguir, ni mucho menos una receta. Se trata más bien de una estructura de relaciones entre los componentes de una propuesta educativa, diseñada y construida deliberadamente a partir de una concepción de educación. En el modelo pedagógico se articulan factores tales como los contenidos, las actividades, las formas de interacción, y los recursos y medios de apoyo, en función de propósitos educativos definidos y mediante la organización de una metodología de trabajo (Turcott, 1999).

Generalizando, un modelo es un esquema que enuncia los elementos de un dominio y las relaciones entre ellos; se basa en un cuerpo de conocimiento definido, en forma de teorías o ideologías al lado de experiencias; pero no ofrece soluciones, sólo una suerte de lo que debería ser. Los modelos no ofrecen el cómo, que es papel de las metodologías y las técnicas, ni el por qué, que lo hace la teoría. Pero ofrece algo que es particularmente útil para el proceso de producción de software: estructuras y relaciones, que pueden transformarse en requerimientos de software.

De modo que para modelar un software educativo, se requiere partir de un modelo pedagógico, que exponga los elementos que intervienen en el proceso educativo. Por ello, en la primera parte de este capítulo se hará un repaso muy sucinto del enfoque pedagógico moderno, para obtener un modelo pedagógico fundamentado. Entonces la búsqueda se afinará comentando aspectos relevantes de la modalidad de educación a distancia, especialmente la que utiliza internet y posteriormente se enfocará aún más en un aspecto esencial para esta investigación: la evaluación del aprendizaje. A continuación se repasará la literatura que trata lo relacionado con los estándares de software, para tener el marco completo que haga posible, en el capítulo siguiente, obtener los requerimientos pedagógicos que debe cumplir un software para evaluación del aprendizaje.

El enfoque pedagógico moderno

Como lo expresa Iyangu (1998), “las teorías de la educación intentan explicar razonadamente cómo debe llevarse a cabo la acción educativa. Son normativas y se configuran a través de un conjunto de principios operativos” (p. 56), estos parten de la concepción que se tiene de la naturaleza humana, de qué se cree en relación con la educación, sus posibilidades y sus límites, y más aún, de los términos en que se concibe el mundo, la vida y el hombre.

Con razón las corrientes pedagógicas han partido de teorías filosóficas y sociológicas, y más recientemente de aportaciones del campo de la psicología. Pero la sola teoría educativa no suele ofrecer mecanismos de aplicación y requiere modelos operativos. El arranque de un modelo operativo sería la definición del objetivo de la educación. Lafourcade (1969) entiende la educación como un proceso sistemático destinado a lograr cambios duraderos y positivos en las conductas de los sujetos sometidos a su influencia, con base en objetivos definidos de modo concreto y preciso. Del Campo (s/f), citando a Piaget, califica algo más definitivamente la clase de cambios deseables en los alumnos al afirmar que

el principal objetivo de la educación es crear hombres que sean capaces de hacer cosas nuevas, no solamente de repetir lo que han hecho otras generaciones; hombres que sean creativos, inventivos y descubridores. El segundo objetivo de la educación es formar mentes que puedan criticar, que puedan verificar y no aceptar todo lo que se les ofrezca.

L'associació per a la Recerca Pedagògica TP (2003) añade un componente moral a la definición del objetivo de la educación: "En esto consiste el hecho humano de la educación, en la formación de la conciencia moral, en la capacidad de discernir entre el bien y el mal". A tono con lo anterior, White (1987) enuncia que

la verdadera educación significa más que la prosecución de determinado curso de estudio. Significa más que una preparación para la vida actual. Abarca todo el ser, y todo el período de la existencia accesible al hombre. Es el desarrollo armonioso de las facultades físicas, mentales y espirituales. (p. 13)

En la actualidad destaca el constructivismo como enfoque pedagógico que describe esos cambios deseables en la educación, cómo ocurren o deben ser producidos dichos cambios, y el papel que juegan los diferentes actores en el aprendizaje.

Constructivismo

Desde la primera mitad del siglo pasado y hasta la década de los 80, la teoría psicológica genética de Piaget ejerció una influencia notable en la educación y numerosas propuestas pedagógicas la tomaban como su fundamento. Sin embargo, actualmente las tendencias se inclinan hacia lo que se conoce como constructivismo, un marco teórico más amplio “en el que confluyen, además de la psicología genética, los aportes de la teoría de Vigotsky y los enfoques socioculturales así como de teorías de la psicología cognitiva” (Carretero y Fairstein, 2001). Esta confluencia de diferentes teorías educativas ha dado lugar a numerosos debates y corrientes constructivistas muy variadas. Los cambios de paradigma en la teoría educativa han hecho que se pase del conductismo al cognitivismo y, más recientemente, al constructivismo (Ríos, 2001).

El principio más importante asociado con las nuevas visiones cognitivas es el reemplazo del acostumbrado énfasis en los hechos aprendidos, el conocimiento y las habilidades básicas por la actitud de centrarse en las habilidades de pensamiento más elevadas. En otras palabras, el contenido es remplazado por el proceso. Los constructivistas creen que la adquisición de las habilidades de pensamiento más elevadas no depende del dominio de las habilidades inferiores ni del conocimiento, por lo que no se necesita diferirlas hasta que éstas hayan sido adquiridas (Cunningham, 1998).

Antes del constructivismo, y durante mucho tiempo, prevaleció el enfoque conductista, según el cual no era posible estudiar científicamente lo que ocurría en la mente, por lo que la educación debía centrarse en los estímulos de entrada, las respuestas de salida y los reforzamientos necesarios para aumentar o disminuir la frecuencia de las respuestas. Pero “con la aceptación del enfoque constructivista ya no se concibe al aprendiz como un receptor pasivo

de información sino como un procesador activo que debe establecer relaciones significativas entre sus conocimientos previos y la nueva información” (Ríos, 2001).

Añade Ríos que el constructivismo tiene su base en las ideas del filósofo alemán Emmanuel Kant, quien afirmaba que todo conocimiento comienza con la experiencia, aunque no todo lo que una persona conoce procede de la experiencia. Sin embargo, a partir de ella una persona puede hacer inferencias, formular hipótesis y hasta elaborar reglas y principios. Por eso el constructivismo concede más importancia al estudiante y busca estimular sus recursos personales, convirtiendo el proceso educativo en un diálogo, más que en un monólogo del educador. Además, busca partir de los conocimientos previos del alumno, de modo que el proceso enseñanza-aprendizaje se ha cumplido cuando el estudiante logra establecer relaciones significativas entre su conocimiento, actitudes y valores con el nuevo conocimiento y las nuevas experiencias.

Una de las implicaciones importantes de estas ideas es el énfasis en las diferencias individuales “por cuanto la misma ‘realidad’ puede tener significados bastante diferentes para distintas personas y aun para las mismas personas en diferentes momentos o contextos”. Al basar el desarrollo tecnológico en estas premisas “el reto, nada trivial”, sigue diciendo Ríos (2001), “es cómo hacer para que el software educativo atienda estos aspectos”.

Enfoque sociocultural. Este componente del constructivismo parte de las ideas del psicólogo ruso Vigotsky, quien destaca la influencia predominante del medio cultural. Su teoría enfatiza tres conceptos fundamentales: la zona de desarrollo potencial, la mediación y el andamiaje:

El concepto de zona de desarrollo potencial constituye la base teórica del principio pedagógico que dice que la única buena enseñanza es la que precede al desarrollo. Rechaza la

perspectiva conductista que espera una respuesta correcta y ofrece una recompensa; antes bien, propone registrar y reconocer los avances del estudiante en una secuencia de ejercicios, tomando en cuenta tanto su punto de partida o nivel de desempeño real, como los recursos empleados: apoyo del sistema y del educador, el tiempo usado, etc.

El segundo concepto fundamental, la mediación, se entiende como “la experiencia de aprendizaje donde un agente mediador, actuando como apoyo, se interpone entre el aprendiz y su entorno para ayudarle a organizar y a desarrollar su sistema de pensamiento y facilitar la aplicación de los nuevos instrumentos intelectuales a los problemas que se le presenten” (Ríos, 2001).

Andamiaje, el tercer concepto, se refiere a graduar la dificultad de la tarea y organizar la ayuda en función de las dificultades del sujeto para enfrentarla. Hay una relación inversamente proporcional entre la ayuda brindada por el educador y la competencia del estudiante. Por tanto, el maestro procura ayudar al alumno a aprender, antes que intentar enseñar. Adoptar este enfoque en la producción de software implica precisamente ofrecer herramientas informáticas que planteen retos a los estudiantes y ofrezcan ayuda de acuerdo con la competencia de cada uno. Esto es similar a la búsqueda del santo grial de las aplicaciones web: la personalización total.

Metacognición y transferencia. Dos elementos determinantes suelen aparecer en los enfoques pedagógicos vigentes: La metacognición y la transferencia de aprendizajes.

El concepto de metacognición se refiere a la conciencia que una persona tiene de las estrategias que aplica en la solución de los problemas, al reconocimiento conciente de la forma en que aprende. La transferencia describe el objetivo final de aplicar lo aprendido en la escuela al medio en el que el alumno se desenvuelve; aunque puede haber algún tipo de transferencia

espontánea “las investigaciones han demostrado que este supuesto es falso y que, por tanto, debe haber una acción deliberada, intencional y sistemática para lograrla adecuadamente” (Ríos, 2001).

Objetos de aprendizaje. Los cognitivistas aportan la idea de que la instrucción puede descomponerse en “objetos de aprendizaje” de uso general. Esta es una idea compatible con la tendencia del software a construirse con componentes encapsulados y autosuficientes para poder ser reutilizados.

En la práctica, así como en la literatura, este concepto es usado abundantemente pero no definido estrictamente. La Learning Technology Standards Committee del Institute of Electrical and Electronics Engineers (IEEE LTSC) ha hecho una propuesta de definición estándar muy amplia: Un objeto de aprendizaje es una entidad, digital o no, que puede ser usada, reutilizada o referenciada en la educación basada en la tecnología; puede tratarse de materiales impresos, tareas de estudio, ejercicios, textos de estudio, casos, recursos mediáticos, cursos, programas de estudio e incluso de personas. La idea fundamental es que un objeto de aprendizaje puede permanecer independiente y ser reutilizado. En la práctica esto significa que la mayoría de los objetos de aprendizaje son pequeños, más pequeños que un curso, y que pueden reutilizarse en diferentes cursos (Koper, 2001, p. 4; véase también IMS Learning Design Information Model, Version 1.0 Final Specification, 2003, p. 17).

Aunque la idea es muy atractiva, no hay consenso entre los educadores acerca de que esto sea posible, pues se piensa que algunos tipos de instrucción no permiten una descomposición de sus elementos para empacarlos en unidades reutilizables. Dice Rodríguez (2001) que una de las respuestas que suelen darse a las posiciones cognitivistas es que las partes no siempre pueden reconstruir el todo. “Tampoco los ‘objetos de aprendizaje’

mínimamente estructurados, no de tamaño mínimo, y creados en un marco cultural y social determinado son fácilmente reutilizables en otros, o ni siquiera es planteable”.

La educación virtual

“Una definición básica de educación a distancia es la entrega del proceso educativo a receptores que no están en la proximidad de la persona o personas que administran o conducen el proceso” (Berge y Collins, 1995, p. 14). La educación a distancia no es un modelo pedagógico sino una estrategia de enseñanza, aunque no puede dudarse que ha tenido su impacto en el replanteamiento educativo que se ha dado en los últimos años, dado que se basa en estimular el autoaprendizaje.

Los sistemas de enseñanza no presencial, denominados originalmente enseñanza por correo y posteriormente enseñanza a distancia y enseñanza abierta, surgieron con la intención de alcanzar a un público que estaba fuera del área de influencia de las instituciones educativas, aunque sin modificar los contenidos y las metodologías usados tradicionalmente. Lo importante era acreditar los cursos mediante exámenes finales, aplicados en las mismas condiciones que experimentaban los alumnos presenciales.

Incluso cuando, a partir de la década del 60 se introdujeron materiales audiovisuales, que luego incluyeron videos educativos, las metodologías siguieron siendo una variación de la modalidad presencial. Lo mismo ocurrió con la incorporación de la internet (Santángelo, 2000).

Introducción de la computadora

El primer uso de las computadoras en instituciones educativas coincide con la aparición de la segunda generación de computadoras a fines de los 50, aunque sus primeros

usos fueron administrativos. Poco después, algunas grandes instituciones empezaron a desarrollar investigación asistidos por computadoras. Una de tales aplicaciones fue el proyecto Plato en la Universidad de Illinois, que comenzó en 1960 con la meta de diseñar un enorme sistema basado en computadoras para la instrucción. Hasta mediados de los 70, estos recursos estuvieron limitados a las universidades de prestigio, unas cuantas corporaciones privadas y el ejército (Alessi y Trollip, 1985).

En el caso particular de México, la educación a distancia ha sido encabezada especialmente por la Secretaría de Educación Pública y la Universidad Nacional Autónoma de México (UNAM); el gobierno federal la formalizó al promulgar la Ley General de Educación en 1956. El ejemplo de la telesecundaria en México es uno de los pocos que ha mantenido un entorno televisivo de educación, habiendo realizado una evolución importante en cuanto a los productos audiovisuales (Santángelo, 2000).

En lo que respecta al uso de internet, la UNAM también ha sido la pionera. Dice Burkle (2000) que hasta febrero de 1998 entre el 50% y el 60% del tráfico de internet en México pasaba a través de un servidor de la UNAM. En esa misma institución se inició el programa de universidad en línea en 1997, mientras que entre las instituciones privadas se destacan el Instituto Tecnológico de Estudios Superiores de Monterrey (TEC) y el Instituto Tecnológico Autónomo de México (ITAM).

En la última década el panorama informático ha cambiado radicalmente e internet se ha convertido en el medio de comunicación estelar de la generación actual. Que la educación la usara como estrategia era sólo cuestión de tiempo.

La internet ha traído consigo innovaciones que han impactado en los conceptos pedagógicos que originaron la educación a distancia, gracias a que ofrece un entorno novedoso

y a que la tecnología permite hoy por hoy ejecutar acciones que antes, aunque se consideraban ideales, eran imprácticas; un ejemplo de ello es la aplicación de exámenes particularizados, o generados al azar y, por lo tanto, diferentes unos de otros. Algunas ventajas que ha ofrecido la internet, que podrían verse como barreras a las que se enfrentaba la educación a distancia clásica, son: (a) Permite mayor interacción entre instructores y estudiantes; (b) el mismo medio permite la conexión con otros recursos, como las instalaciones escolares, del trabajo o del hogar; (c) ayuda a que el nivel de lectura y escritura se corresponda más con el nivel de instrucción; y (d) aumenta las oportunidades de autodirigirse y autoadministrarse en el desarrollo personal (Berge y Collins, 1995).

Internet se ha negado a quedar reducida a una herramienta didáctica más y ha reclamado su propio espacio. Hoy se habla llanamente de educación a distancia a través de internet, educación telemática, educación virtual o digital, teleformación, elearning o educación en línea. Son términos equivalentes con toda una carga sobreentendida (Picardo, mayo de 2002). Este nuevo ambiente, que transformó la educación a distancia en educación virtual, se ha idealizado como un conjunto de espacios educativos interactivos, flexibles, sincrónicos y asincrónicos, cooperativos y personalizados (Ramírez, 2002).

Aquí conviene precisar que no todo proceso educativo que utiliza la computadora o internet es educación virtual. Boettcher (Citada en College of Continuing Education, 2002) distingue más bien cuatro tipos de cursos en lo que respecta a su actividad web:

1. Los cursos presenciales que utilizan internet para publicar información general
2. Los cursos presenciales auxiliados por la web: utilizan internet para distribuir materiales de clase

3. Los cursos mixtos: siguen teniendo componentes presenciales, pero el peso principal recae en internet; y

4. Cursos en línea: propiamente educación a distancia mediante internet, carentes de elementos presenciales.

Adaptación al nuevo entorno

No obstante el sobreentendido con que se asume cotidianamente la educación virtual, también en ella lo esencial es el modelo pedagógico. Para Picardo (mayo de 2002), este modelo es el resultado de establecer una plataforma web sobre la que se coloquen los espacios tradicionales de aulas y bibliotecas virtuales, y que, especialmente “cuente con diseños de guías didácticas para la orientación que eviten el ‘naufragio’ en la navegación, y que dichas guías estén sustentadas en una concepción psicopedagógica coherente y en un manejo versátil, a través de tutores especializados”.

Ahora bien, reformulando la pregunta que abre este capítulo, ¿por qué es obligado definir el modelo pedagógico cuando se habla de aplicar las nuevas tecnologías de la información a la educación virtual? Santángelo (2000) ofrece las siguientes razones y problemas que hacen imprescindible definir e implementar un modelo pedagógico por encima de las herramientas tecnológicas:

1. El rápido avance tecnológico está relegando el sustento de la educación, los modelos pedagógicos, a costa de enfatizar el medio.

2. En la educación virtual la asincronía y el incremento de la distancia física plantean desafíos en el terreno de la interacción y la comunicación, que los modelos presenciales no contemplan y que las herramientas sólo resuelven en el ámbito instrumental.

3. Es necesario cuestionar ventajas que ofrece la tecnología que se dan por verdaderas, como la posibilidad de llegar a un gran número de estudiantes, mientras se reducen los costos y se requieren menos docentes con menor preparación didáctica.

4. Existe la tendencia a colocar en el centro del proceso educativo a los contenidos y a las herramientas informáticas, desplazando los procesos de interacción y construcción de significados compartidos entre docentes y estudiantes.

5. Hay el peligro de adoptar un modelo economicista, en detrimento de la pertinencia e importancia de los modelos educativos.

6. Y finalmente, los procesos de enseñanza y aprendizaje que tienen lugar en el universo virtual, como en la educación presencial, también deben derivar y ser coherentes con una concepción sicopedagógica y social que guíe el enfoque educativo, los programas, las situaciones y actividades, etc.

A pesar de la claridad de estas razones, no ha sido tarea fácil para la pedagogía recuperar su necesario protagonismo en la educación virtual. Con frecuencia se ha debido a discusiones sobre cuestiones fundamentales. Para empezar, ¿existe tal cosa como un modelo para el elearning? Y es que la educación virtual es difícil de caracterizar debido a su relativamente corta vida y a las diversas maneras de implementarla. Es cierto que la mayoría de los programas virtuales reflejan las características de la educación a distancia, como la masificación, programas de estudio más estructurados, industrialización del desarrollo de cursos y división del trabajo pedagógico, pérdida de oportunidades de aprendizaje informales, etc., pero es innegable que posee otras muy particulares, como requerimientos técnicos definidos, necesidad de ciertas habilidades y el uso integrado y dinámico de muy diferentes medios (Oliver, 2000).

La discusión en los círculos educativos también se ha centrado en definir si la educación en línea es medio o fin, inclinándose hasta ahora el consenso por una visión instrumental de la telemática. Esta solución, sin embargo, no impidió otra discusión más: ¿Con qué enfoque es compatible la educación virtual? Según Picardo (mayo de 2002), con prácticamente todas las que rigen actualmente: Con el constructivismo y los tres elementos fundamentales de toda situación de aprendizaje: Contenidos, procesos y condiciones; con la teoría de la Conversación de Pask, que supone que aprender es por naturaleza un fenómeno social; con la teoría del conocimiento situado de Young, la cual señala que el conocimiento es una relación activa entre el individuo y un determinado entorno, y que el aprendizaje se produce cuando el aprendiz está envuelto activamente en un contexto complejo y real; y con la teoría de acción comunicativa de Habermas, sustentada en el rigor, la racionalidad y la crítica.

También para Cardona (2002) la educación virtual encaja perfectamente en las teorías sicopedagógicas más relevantes de hoy en día. Las características de internet y su potencialidad son en muchos modos un eco de esas propuestas educativas. En cuanto al constructivismo dice que “la internet presenta rasgos de un entorno de aprendizaje constructivo... Es un sistema abierto guiado por el interés, iniciado por el aprendiz, e intelectual y conceptualmente provocador” (p. 9).

Justifica igualmente la relación con la segunda teoría más frecuentemente invocada, después del constructivismo, para fundamentar la validez de la internet como vehículo educativo, a saber, la teoría de la conversación, basada en las ideas de Vigotsky. “La internet [se] adhiere a la noción vigotskiana de interacción entre gente que trae diferentes niveles de experiencia a una cultura tecnológica. La internet es un entorno que presupone una naturaleza

social específica y un proceso a través del cual los aprendices crean una zona virtual de ‘proximal development’” (Cardona, 2002, p. 9-10).

Y finalmente, como Picardo, citado más arriba, Cardona agrega una tercera teoría a la que suele acudir al hablar de educación virtual: la teoría del conocimiento situado. Según este autor, la internet cumple las dos premisas de esta teoría: realismo y complejidad. Por un lado, la internet posibilita intercambios auténticos entre usuarios provenientes de contextos culturales diferentes pero con intereses similares. Por otro lado, la naturaleza inestable del entorno Internet constituye un escollo para los no iniciados, que sin embargo, y gracias a su participación periférica continuada, se ven recompensados con una enculturación gradual.

Modelos para la educación virtual

Por lo que se ve, internet se revela como un medio muy plástico, ya que la estrategia de la educación en línea ha dado lugar a la formulación de modelos derivados, es decir, modelos pedagógicos pensados especialmente para el ambiente informático. Por ejemplo, Gómez, Herrera y Bermúdez (2001) proponen un modelo para cursos de posgrado que “asume una organización curricular por módulos y unidades didácticas, es decir, diseñada alrededor de un problema”. Este modelo es descrito como sigue:

1. Uso de métodos y técnicas que propicien la interacción entre todos los actores del curso.
2. Evaluación integral, del diseño curricular, del diseño informático, activa, que permita la auto y la coevaluación.
3. Estructurado en tres secciones: Introdutoria, para informar de las finalidades del curso, cuyo propósito sea motivar; de desarrollo, organizada en módulos y unidades didácticas que presenten no el contenido total sino la literatura básica y actividades de aprendizaje; y de

cierre, que propone actividades integradoras basadas en problemas reales en el contexto de la actuación profesional de los estudiantes.

Las autoras hacen esta propuesta porque creen que en internet es posible, aunque eso no quiere decir que sea automático que agregar tecnología informática a la educación produzca modelos pedagógicos nuevos. No se trata sólo de agregar nueva tecnología a las viejas maneras de organizar la enseñanza y el aprendizaje, “el cambio de paradigma es ir de un ambiente de enseñanza a un ambiente de aprendizaje” (Berge y Collins, 1995, p. 4). La base de esta afirmación descansa en el reconocimiento formal de que hay diferentes estilos de aprendizaje, que los estudiantes crean su propio significado cuando aprenden nuevas cosas, y que lo que hace la diferencia en la retención y transferencia del contenido no es tanto lo que hacen los maestros, sino lo que los estudiantes como sujetos que aprenden pueden animarse a hacer por ellos mismos. Estos conceptos están alineados con las capacidades de autogestión que ofrece la internet y con la tendencia a ceder a los alumnos una parte sustancial de la responsabilidad de su propia educación.

Otra propuesta es el modelo de Misión infinita, llamado así porque su esquema recuerda el símbolo matemático ∞ (Figura 1); “creado para guiar el diseño y desarrollo del currículo y la instrucción en línea, el modelo inicia y concluye con la misión”. En el esquema resultan evidentes dos procesos paralelos: pensar y hacer, los cuales nunca deben estar separados del todo; y también destaca la naturaleza continua del desarrollo (Siebold y Jeffery, 2003, p. 5).

A diferencia de los dos ejemplos mencionados, muchas formas de educación virtual trabajan sólo en torno a la creación de contenidos y a la migración pasiva a internet de las formas tradicionales de enseñar y aprender (Stacey, octubre de 2003). Pero se requiere un

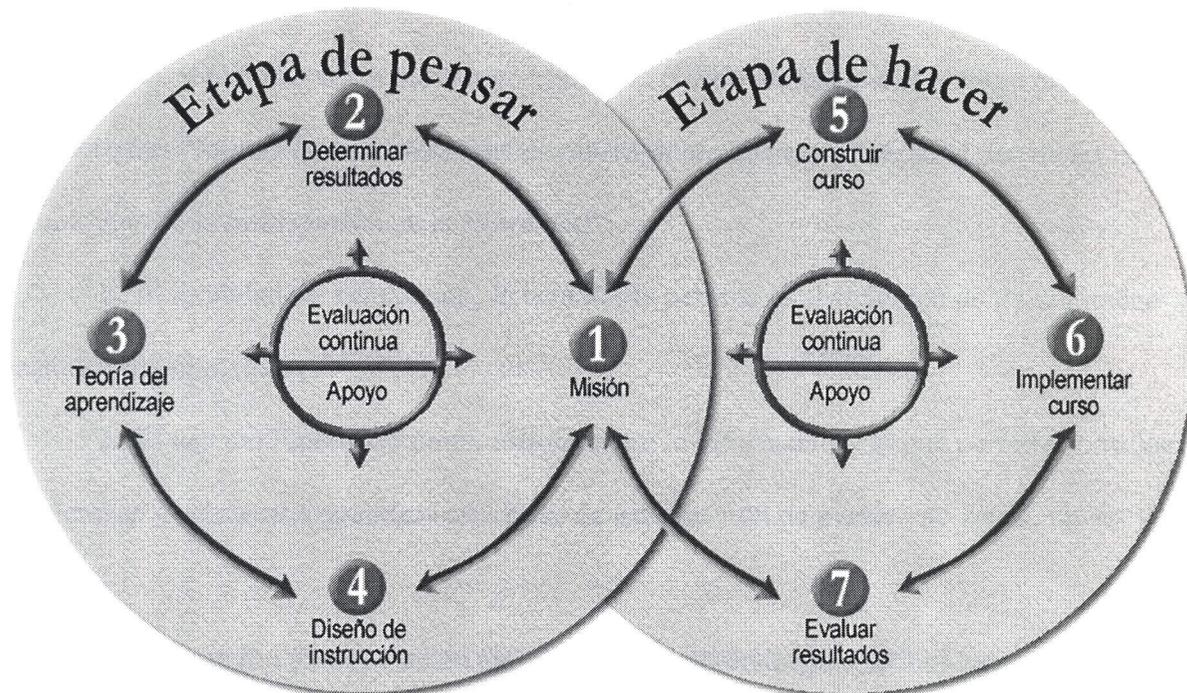


Figura 1. Modelo de misión infinita creado para guiar el diseño y desarrollo del currículo y la instrucción en línea

cambio más que cosmético. “Estos escenarios demandan una nueva arquitectura educativa que apunte y apueste al aprendizaje de por vida (lifelong learning)”; se trata de llevar a la práctica la hipótesis de enseñar a aprender (Picardo, mayo de 2002).

El propio Picardo enuncia un modelo pedagógico más abstracto, para sustentar la educación virtual, llamado “pedagogía informacional, ante el cual, los docentes y estudiantes deben asumir un nuevo rol de ‘mediaciones’ entre la experiencia humana y la información existente, y sobre todo caer en la cuenta que la información debe ser punto de partida y de llegada en el proceso de enseñanza aprendizaje”. Si internet ha puesto de relieve el poder de la información y el conocimiento, resulta natural que la pedagogía sea invadida por este nuevo paradigma.

Pero con estos y otros modelos ¿realmente la educación virtual es capaz de satisfacer los principios de la educación moderna? Mancinas (1999) afirma que sí y como prueba enlista las diferentes “formas de aproximación al conocimiento que caracterizan el uso de las nuevas tecnologías de la información en la educación”:

1. El modelado y simulación; la tecnología permite la abstracción de un fenómeno y su posterior manipulación;
2. El uso del hipertexto como estructura de la información, lo que permite consultar, seleccionar y estructurar grandes cantidades de información de manera no lineal, rápida y eficiente;
3. Interacción y cooperación entre grupos de aprendizaje distribuidos;
4. Interacción con el entorno físico, lo cual puede lograrse acoplando sensores a los dispositivos computacionales y a las redes informáticas;
5. Incorporación del arte: música, pintura, cine, video, multimedia u otras formas de expresión en formato digital; en lo que toca a multimedia Stacey (2003) es particularmente optimista en cuanto a la incorporación de audio al elearning, debido a que la tecnología lo ha facilitado enormemente;
6. Acceso personalizado y a distancia de la información;
7. Aprendizaje cooperativo; y
8. Formación de comunidades de aprendizaje.

Berge y Collins (1995) y Campbell y Pargas (2003) ofrecen por su parte opciones prácticas y sencillas con las que la educación cuenta ahora merced al uso de la informática, opciones que al mismo tiempo estimulan el uso de la tecnología y obligan a replantear los supuestos pedagógicos:

1. Mantener accesible material de clase en internet;
2. Llevar a cabo ejercicios de aprendizaje colaborativos aprovechando las facilidades de comunicación instantánea de las computadoras conectadas a una red;
3. Ofrecer realimentación inmediata, por ejemplo, aplicando encuestas rápidas, o utilizando herramientas de exámenes que ofrezcan resultados en cuanto el ejercicio concluya;
4. Discusiones dirigidas en grupo;
5. Lluvia de ideas en foros en línea;
6. Ponencias no presenciales;
7. Discusión en pequeños grupos vía electrónica; etc.

Por su parte Stacey (octubre de 2003) enfatiza esos recursos propios de internet que han llegado o pueden llegar a formar parte indiscutible del ambiente educativo:

1. Foros de discusión
2. Blogs: Diarios personales en formato electrónico
3. Encuestas rápidas. Su facilidad de implementación permite realimentación inmediata sobre casi cualquier asunto.
4. Webquest: Son un tipo complejo de actividad, orientado hacia la investigación, que se compone de un conjunto de tareas o desafíos, recursos y ayudas y un esquema de evaluación. Estos esquemas, conocidos como rúbricas, suelen ser multidimensionales. Lo particular de esta actividad es que todos los recursos con los que interactúan los alumnos están en internet (Dodge).
5. Juegos y simulaciones

6. Webcast: Todo el cúmulo de transmisiones multimedia que existe en el ciberespacio: Audio y video, voz por IP, pizarras interactivas, tableros de discusión, radio, chats, etc.

Puntos álgidos del elearning

Quizás porque hoy es más fácil que nunca antes alzar (digitalmente) la voz para opinar, el caso es que internet y la educación virtual dan mucho de que hablar cada día. Y no son necesariamente buenas noticias para todos; basten como ejemplos estos temas que están sobre la mesa:

La vigencia de las instituciones en duda. En el campo de las organizaciones, internet comienza también a plantear cuestiones que antes podrían considerarse sacrílegas, como la relacionada con la vigencia de las instituciones educativas formales.

Afirma Collazos (2000) que “ya no va a ser necesario atiborrar al estudiante de pregrado de conocimientos ‘por si acaso los necesita’ (*just in case*). Se impone más bien una excelente formación básica en el pregrado y una formación especializada cuando se necesite (*just in time*)”. Otra variedad la constituiría la educación “a la carta”, programas académicos hechos a la medida de cada estudiante. El uso de la tecnología web ha permitido que estos casos sean atendidos por organismos no educativos, como empresas capacitadoras o instituciones certificadoras.

“Estamos acostumbrados a delegar en los centros de educación superior la certificación de la calidad de los profesionales”, continúa Collazos. “Pero, ¿seguirá siendo así en el futuro? Todo parece indicar que no”. Por otro lado, dado que se ha estado reconociendo la necesidad de implantar un proceso educativo que abarque toda la vida, mediante estrategias de educación

continúa, ésta “podría llegar a ser más importante (al menos si la medimos por el tiempo que la persona le dedica) que la formación de pregrado lo cual contribuiría a erosionar el papel certificador de la universidad” (Collazos, 2000).

Globalización educativa. La demanda educativa parece aumentar debido a que la tecnología la ha masificado, al menos potencialmente. Pero como satisfactores entran no sólo entidades educativas públicas y privadas, sino organizaciones comerciales, empresas capacitadoras, proveedores de servicios, etc. Stacey (diciembre de 2003) considera que estos actores necesitan dialogar acerca de este viraje educativo, y responder preguntas como: ¿Es la educación una mercancía para lucrar o un servicio público? ¿Globalizar la educación es sólo un asunto de distribución de contenidos o es obligado considerar aspectos de diversidad, entornos sociales e intercambio cultural?

Derechos sobre el contenido. También Stacey (diciembre de 2003) abunda en un tema que en la industria discográfica ha polarizado a los participantes y que puede trasladarse al contexto educativo. El punto es quién tiene derechos sobre el contenido y cómo controlar esos derechos sin dificultar el intercambio en el ambiente de aprendizaje. En un extremo se encuentran quienes proponen que prácticamente todo esté regulado y que todos los derechos estén reservados, mientras que en el otro extremo se ubican los defensores de soluciones más anárquicas, como los movimientos de software libre y del *open source*. El autor agrega que una solución intermedia sería crear mecanismos que permitan al autor ser quien especifique acciones como decidir quién y cómo copia sus contenidos, la distribución, el uso, etc.

En suma, como puede verse hasta aquí, el hecho de considerar a la educación virtual como una estrategia o un medio, no impide que siga trayendo a colación conceptos,

metodologías e implicaciones que están lejos de definirse satisfactoriamente. Podría bien elegirse un modelo pedagógico basado, por ejemplo, en el constructivismo, pero eso no terminaría la tarea; todavía sería necesario particularizarlo tomando en cuenta las oportunidades que el ciberespacio ofrece.

Evaluación del aprendizaje

De todo el proceso educativo interesa una de sus fases culminantes, que no última, si se ha de creer en la educación continua, y también si se ha de considerar, como se expondrá más adelante, que el fin no es evaluar ni obtener una nota, sino confirmar que se ha aprendido, verificando el nivel de rendimiento del individuo respecto a un dominio bien definido de conductas identificadas de un modo simple y de conveniencia con el conocimiento (Álvarez, 2001, Lafourcade, 1969). Esta fase es el programa de evaluación del aprendizaje o evaluación instruccional, “que comprende la valoración de los logros de los estudiantes, de la ejecución del maestro y de la efectividad de las metodologías y materiales utilizados” (Glabán y Ortega, 2002), si bien esta investigación se limitará a lo relativo al desempeño estudiantil.

Debe quedar claro desde el principio que evaluar es más que medir; es interpretar una medida en relación con una norma preestablecida (Cunningham, 1998; Wiggins, 1998). Por lo tanto, evaluar tampoco es igual a calificar (véase Pérez, 1997), porque podría incluso evaluarse el desempeño o un producto sin asignar una calificación.

Lamentablemente ponerse de acuerdo en el significado del término no elimina las dificultades que encierra la evaluación, toda vez que lo que debe medirse para empezar son el rendimiento escolar, y conductas y actitudes en los alumnos. Además, el propio hecho de evaluar influye en los evaluados y distorsiona los resultados, sin contar que la experiencia de

evaluación es, estrictamente hablando, irrepetible, lo que dificulta el hacer comparaciones absolutas entre resultados (Pérez, 1997; véase Lafourcade, 1969).

¿Cómo puede definirse entonces el campo de evaluación? Es decir, ¿cómo determinar cuáles aspectos del aprendizaje pueden ser sujetos de una evaluación que resulte significativa y útil? Lafourcade (1969) propone las siguientes condiciones que deberán reunir los rasgos o características de la conducta sujeta a medición: ser comunes a un grupo de personas, definibles con claridad y precisión, variables en relación con los fenómenos que las detentan y que promuevan reacciones muy similares.

Crítica a los exámenes tradicionales

Aunque se acepta que evaluar es parte sustancial del proceso educativo, no todos concuerdan con la forma de evaluar. En lo que parece estarse produciendo un acuerdo casi generalizado es en rechazar los métodos tradicionales de evaluación, con los cuales se evalúa sólo al alumno, pasándose por alto la instrucción o al maestro, que se centra en los resultados, considera sólo los efectos observables y no contextualiza, eso sin considerar su uso como instrumento de poder. La herencia de esta perspectiva es la tendencia a sólo buscar que la evaluación sea confiable y válida, por lo que su base teórica se ha centrado en el instrumento, es decir, en la medición, dejando de lado factores fundamentales en torno al objeto de evaluación (Picardo, mayo de 2002).

Pero hay otro extremo, según lo expresa Cunningham (1998), ya que la única forma de evaluación que los constructivistas parecen apoyar es la comunicación personal con los estudiantes para mejorar la instrucción, porque creen que el foco primario de la evaluación es el estudiante y que las consideraciones de calificación deben dejarse de lado. La evaluación,

entonces, debe ser auténtica, lo cual significa que “debe ser realista, significativa y relevante para una comunidad práctica, compleja y rica en información” (Stacey, octubre de 2003).

Sin embargo, en la práctica los constructivistas han aceptado el hecho de que las evaluaciones no serán eliminadas por completo, así que al menos procuran que se adopten metodologías más apropiadas, como el reemplazo de los exámenes objetivos convencionales por la evaluación del desempeño, o presentación de portafolios.

Abundando en la crítica a la evaluación tradicional, y particularmente en lo que se refiere a los exámenes, para Díaz (1993) llama la atención “la excesiva confianza (social, política e individual) que pretende resolver los problemas de la educación sólo modificando los exámenes y el reduccionismo con el que se aborda el problema de la evaluación, que hace de este tema sólo una cuestión estadística, desconociendo otras dimensiones de la problemática” (p. 7). Agrega que ha ocurrido una involución pedagógica en este terreno, pues según la evidencia histórica antes de la pedagogía moderna el examen formaba parte del método; era “un lugar de aprendizaje, no de verificación... era un espacio público para mostrar la competencia que se había adquirido”.

Pero hoy en día, dice Picardo (mayo de 2002), el modelo pedagógico “exige centrarse en la ‘valuación’, en los juicios de valor, lo cual es más amplio y considera otras variables menos visibles pero existentes circunscritas a la realidad a ser evaluada”. El problema es, continúa este autor, que generalmente se mide y sobre los datos estadísticos se sacan conclusiones, pero no se toma en cuenta el entorno inmediato que determina a los alumnos o escuelas que se evalúan; por ejemplo, se ignoran por completo variables como la madurez, la responsabilidad o la integridad.

Clases de evaluación

Conforme las críticas han ido generando nuevas soluciones, ha ido quedando en claro que, independientemente de los métodos que se adopten, ocurren diferentes clases de evaluación, según su objetivo y la forma de aplicación (Tabla 1). No que sean evaluaciones diferentes, sino la evaluación del aprendizaje tiene diferentes momentos y funciones (Pérez, 1997).

Tabla 1
Clases de evaluación

según su objetivo	según su aplicación
Formativa	Formal
Sumativa	Informal
Diagnóstica	Alternativa

La evaluación formativa la componen las diferentes actividades que tienen el propósito de proveer realimentación a maestros y alumnos. Suelen cubrir pequeñas partes del curso, ser periódicas, no durar mucho tiempo y enfocarse en contenidos muy específicos (Pérez, 1997; Bloom, Madaus y Hastings, 1981). La intención es descubrir cuánto dominan los estudiantes los contenidos recién expuestos con el fin de afirmar o corregir el proceso.

La evaluación sumativa, en cambio, está orientada a una valoración más general, abarcando todo o grandes partes del curso; pretende describir objetivamente, mediante una calificación, el dominio de un conjunto de habilidades o conceptos (Pérez, 1997; Bloom, Madaus y Hastings, 1981). Para esta clase de evaluación, especialmente en el ambiente de internet, parece más apropiado utilizar la presentación de portafolios, que es definido tras una discusión entre maestro y alumno para determinar los criterios de prueba y qué debe incluir el portafolio, por ejemplo: exámenes objetivos y basados en casos, trabajo de equipo, revisión de

trabajos por colegas, actividades de discusión, elaboración de ensayos, estudio de casos, reunión de recursos web, etc. (College of Continuing Education, 2002).

El caso especial, diferente de las anteriores y de menos uso, es la evaluación diagnóstica (Pérez, 1997). Sus defensores afirman que es inapropiado asumir, como es común que muchos docentes lo hagan, que al inicio de un curso todos los estudiantes tienen cero y deben ir ganando su 100, dado que no hay un trasfondo educativo homogéneo en todos ellos. Por eso es necesaria una evaluación inicial, para descubrir si un estudiante está abajo o arriba de ese cero hipotético. Este dato será útil cuando la calificación final se construya tomando en cuenta el grado de avance que tuvo un alumno respecto a su punto de partida. También conviene la evaluación diagnóstica, llevada a cabo una vez, antes de iniciar formalmente el curso, para verificar el dominio que se tiene de habilidades o conocimientos que son considerados prerrequisitos. Y una tercera razón para usarla sería el clasificar a los estudiantes de acuerdo con ciertas características (Bloom, Madaus y Hastings, 1981).

Si bien Cunningham (1998) sostiene también la división de la evaluación en formativa y sumativa, en lo que concierne a su aplicación ofrece otra forma de dividirla: Formal, informal y alternativa. La primera se compone de los métodos y exámenes convencionales y se caracterizan por su precisión y exactitud. La evaluación informal incluye cuestionar directamente a los estudiantes, observarlos al realizar sus tareas, proponerles ejercicios de articulación de problemas, etc.; por supuesto, se trata de una evaluación totalmente subjetiva e imprecisa.

Evaluación alternativa

La evaluación alternativa, también conocida como evaluación auténtica, es un punto intermedio entre las anteriores. Es una estrategia usada tradicionalmente en la educación

artística, donde es difícil aplicar exámenes y tareas objetivos, pero últimamente ha invadido otros terrenos, pues su idea central es ir más allá de las técnicas usuales de exámenes de falso-verdadero, opción múltiple, etc., para incluir exámenes de desempeño y portafolios. Carece de la precisión de la evaluación formal, pero no es tan subjetiva como la informal.

Aunque se plantea en términos algo absolutos o ideales, puede ser clarificadora la comparación que Wiggins (1999) hace entre la evaluación alternativa y la tradicional, según se presenta en la Tabla 2.

Tabla 2
Comparación entre la evaluación alternativa y la evaluación tradicional

evaluación alternativa	evaluación tradicional
Examina directamente el desempeño en tareas intelectualmente significativas	Se basa en ítems obtenidos indirectamente a partir de los cuales se hacen inferencias acerca del desempeño
Requiere que los estudiantes manejen efectivamente el conocimiento adquirido	Tiende a revelar sólo si el estudiante puede reconocer o recordar lo que aprendió fuera de contexto
Las tareas reflejan las prioridades y desafíos en las mejores actividades instruccionales	Las tareas se limitan normalmente a evaluaciones en papel con cuestiones de una sola respuesta válida
La validez y confiabilidad descansan en estandarizar criterios para valorar una variedad de productos	Se basa en la estandarización de ítems; es decir, en definir la respuesta correcta
Sus resultados se proyectan hacia las fases del aprendizaje que siguen	Suele limitarse a ser un reflejo de una enseñanza previa

Lo que la mayoría de los defensores de las actividades de evaluación tradicionales no pueden ver, dice Wiggins (1999) es que es la forma, no el contenido de la actividad, lo que perjudica el aprendizaje. El verdadero desafío es ir más allá de demostrar una validez técnica estándar. Entonces, ¿qué cualidades deben procurarse según este autor para una actividad de evaluación auténtica?

1. Que sea realista
2. Que requiera juicio e innovación
3. Que impulse a los estudiantes a realizar el cometido propuesto
4. Que replique o simule el contexto en el que los adultos son “probados” para el trabajo, la vida civil y la vida personal
5. Que evalúe la habilidad del estudiante para usar eficiente y eficazmente su repertorio de conocimiento y habilidades para enfrentarse a una tarea compleja
6. Que permita las oportunidades apropiadas para ensayar, practicar, buscar recursos, recibir realimentación y afinar el desempeño y los resultados

Métodos de cálculo de la calificación

Si bien la tendencia es considerar que la verdadera evaluación no tiene como fin obtener una calificación, dicha nota se busca obtener como una forma de objetivizar la medida en la que el estudiante ha hecho suyos los contenidos. El problema radica en encontrar el método o conjunto de métodos que logre esto con suficiente precisión y objetividad, y sea al mismo tiempo práctico y factible. Otro problema es que muchos maestros carecen de las habilidades necesarias para aplicar efectivamente diferentes técnicas de evaluación, dado que el fundamento de medición de estas técnicas es matemático. Incluso en el caso en el que se dominen esas bases matemáticas, llevar a cabo el proceso manualmente consume mucho tiempo (Cunningham, 1998).

La forma tradicional de calificar, y la más fácil, consiste en asignar notas absolutas, pero se han intentado otras fórmulas. La idea básica de algunas de ellas es calificar en función del rendimiento escolar del grupo; es decir, calificar a un alumno relacionando su desempeño con el de los demás. Algunas técnicas son grados equivalentes, rangos percentiles y las

puntuaciones estándar, como las puntuaciones z y los stanines. Estas últimas están determinadas por un promedio fijo y la desviación estándar; esto es importante, porque los resultados que han sido transformados en una misma escala pueden ser comparados y combinados. Sin embargo, tienen la desventaja de ser más difíciles de entender y aplicar.

Se detallan más adelante las técnicas enlistadas en la tabla 3 (véanse descripciones más completas en Cunningham, 1998).

Tabla 3
Métodos de cálculo de la calificación

Método	resultado obtenido
Notas absolutas	Calificación bruta basada en un 100 arbitrario
Grados equivalentes	Suma del equivalente en puntos decimales, de los aciertos obtenidos en una evaluación estándar para un conjunto de grupos de diferentes grados
Rangos percentiles	Posición ocupada, en porcentaje, en una lista ascendente por alumnos de un mismo grado sometidos a una misma evaluación.
Puntuaciones z	Puntuación bruta en relación con la desviación estándar
T-score	Transformada de las puntuaciones z para mejor lectura = $10z + 50$
Rúbricas	Suma de puntos subjetivos obtenidos en un conjunto de conductas observadas durante una actividad de evaluación compleja. Usada en evaluación alternativa

Grados equivalentes. Es un método simple que otorga a los estudiantes un grado decimal por cada puntuación bruta positiva obtenida en un examen que sirve de base para un conjunto de grupos. Su simplicidad puede ser engañosa porque los grados equivalentes son imprecisos y ofrecen una descripción distorsionada del desempeño de un estudiante, ya que no es un método práctico de compararlo con el desempeño de otros estudiantes en otros grupos y grados.

Rangos percentiles. Es un método algo mejor que el anterior, porque la comparación se realiza entre estudiantes del mismo grado y expresa qué porcentaje de individuos (o casos) se hallan en un determinado puntaje. Simplemente se ordenan ascendentemente los resultados de todos, se divide cada uno entre el número de alumnos y se multiplica el resultado por 100. De este modo todos los resultados están, como porcentajes, entre 1 y 100.

Este método suele usarse al aplicarse pruebas estandarizadas y rara vez en el ámbito de una clase. Lamentablemente se pierde información al hacer esta conversión, porque la evaluación se basa más en la posición que ocupa el estudiante en la lista que en su resultado real.

Puntuaciones z. Algunos lo consideran uno de los mejores procedimientos porque se basa en medidas estadísticas. Suelen ser utilizadas con frecuencia para comparar rendimientos de un alumno en varias asignaturas o entre sus propios compañeros, pero no se recomienda para grupos reducidos; menores a 30, por ejemplo.

Tiene dos desventajas: Cuando la puntuación bruta es menor que el promedio se obtiene un resultado negativo, y usualmente es reportado como un decimal. Para evitar estos problemas, la puntuación z comúnmente es transformada en una puntuación estándar con un promedio de 50 y una desviación estándar de 10; el resultado suele conocerse como T-score: $T\text{-score} = 10z + 50$. La forma de obtenerlas es aplicando $z = (X-M)/\sigma$, donde $\sigma = \sqrt{(\sum X^2)/N}$, M = media, X = puntuaciones brutas y N = número de casos.

Una interpretación de z es la siguiente: Si los valores caen dentro de +1 y -1, z representa el promedio. Si z es mayor que +1, significa muy bueno. Si z es menor que -1, significa regular. Si z es mayor que +2, el sujeto ha cumplido una actuación extraordinaria. Si z es menor que -2, la actuación habrá sido deficiente.

Rúbricas. Esta es una solución ofrecida para la evaluación alternativa y que suele utilizarse en los webquest (Dodge). Se aplica para evaluar actividades complejas, durante las cuales se consideran varias conductas que se pretenden desarrollar en el curso. Una vez definidas las conductas a observar, se crea una tabla con columnas que expresan diferentes grados de éxito; por ejemplo, regular, bien y excelente. Entonces, en la línea de cada conducta se asignan puntos a las columnas. La evaluación finalmente consistirá en ubicar el desempeño del alumno en las cuadrículas correspondientes y sumar (Tabla 4).

Como se observa, ponderar las evaluaciones para obtener calificaciones válidas puede resultar ser una tarea complicada, especialmente porque la exigencia es que el producto obtenido refleje fielmente la realidad.

Tabla 4
Ejemplo de rúbrica

conducta	*	**	***	****
Seguir indicaciones	0 puntos No les presté atención	1 puntos Las seguí pero tenían que recordarme que volviera a mi tarea	2 puntos Seguí las indicaciones y persistí en mi tarea	4 puntos Seguí las indicaciones y ayudé a mis compañeros que no sabían qué hacer
Cooperación con los compañeros	0 puntos No nos pusimos de acuerdo y sólo perdimos el tiempo	2 puntos Pudimos trabajar juntos	3 puntos Hicimos el trabajo juntos pero eludimos la responsabilidad de presentarlo	5 puntos Hicimos el trabajo juntos y asumimos la responsabilidad de presentarlo
Uso de recursos	0 puntos Fui incapaz de encontrar la información que buscaba	1 puntos Encontré sólo una parte de la información	2 puntos Encontré toda la información necesaria, con ayuda de otros	3 puntos Encontré toda la información necesaria de manera independiente
totales				0/12

Como la cuestión del método ya tiene tiempo en el ámbito educativo, siguen vigentes las medidas que recomienda Lafourcade (1969) para obtener notas más significativas, independientemente del método aplicado:

1. Seleccionar un método de conversión de puntajes brutos (extraídos de las diversas fuentes) en notas que representen la situación más real posible de los alumnos, frente al grupo escolar al cual pertenecen.
2. Discriminar las fuentes de información del rendimiento escolar y adjudicarles la ponderación que se juzgue más conveniente reduciéndolas a una sola nota compuesta.
3. Controlar la aplicación del sistema de calificaciones adoptado, a fin de que no aparezcan tantos sistemas como “intérpretes” existan.

Autoevaluación y realimentación

Una consecuencia de los enfoques pedagógicos actuales, es que se ha concedido más peso a la participación del alumno en su experiencia de aprendizaje. Eso incluye la evaluación. “Todos los estudiantes pueden autoevaluar su trabajo con precisión, independientemente de su habilidad para el desempeño” (Wiggins, 1998, p. 2). Por extensión, también la coevaluación es plausible.

Con el fin de que esto sea una realidad, cuatro cosas deben ocurrir a favor de los estudiantes, dice Wiggins (p. 21):

1. Necesitan una completa desmitificación de los exámenes estándar y de desempeño con los que son evaluados, tal como ocurre en la vida profesional cuando se evalúa el trabajo, la interpretación musical o el resultado en un deporte;

2. Necesitan oportunidades múltiples, acompañadas de realimentación, para poder dominar tareas complejas;

3. Necesitan reportes de su progreso, en los cuales su desempeño actual es comparado con el desempeño ideal; y sobre todo

4. Necesitan saber cómo lo están haciendo mientras lo hacen.

“La prueba que está diseñada para enseñar y mejorar, no sólo para medir, es la esencia de la evaluación educativa. Pero, ¿qué significa esto en la práctica? Dos elementos fundamentales: tareas auténticas y realimentación del desempeño amigable e integrada” (p. 22-24).

La realimentación implicada (*feedback*), necesita ocurrir en dos momentos: no sólo después de la tarea, sino durante las tareas de evaluación. Se trata de informar lo que una persona hizo en relación con lo que intentó; en sentido estricto, es lo que confirma o rechaza lo correcto de las acciones. La mejor realimentación es altamente específica y descriptiva del resultado, clara para el estudiante y ofrecida en términos de objetivos o estándares específicos (Wiggins, 1998).

Evaluación en línea

Llevar la evaluación a internet ha seguido la misma ruta que la educación en su conjunto. Algunos procesos y formas no se diferencian de la experiencia presencial; por ejemplo, en la educación virtual el diseño de un examen y el desarrollo de las preguntas es esencialmente igual que en los exámenes tradicionales; la responsabilidad de producir un buen examen, útil y válido, sigue recayendo en el docente (Gail, 2003; véase también Santana, 2002). En realidad los modelos pedagógicos modernos han impactado al proceso educativo independientemente de su modalidad; la incorporación de la internet sólo ha brindado las

herramientas que esos modelos necesitaban para expandirse vertiginosamente y acercarse a su verdadero potencial

La evaluación no es la excepción. Woit y Mason (2003) reportan que su investigación muestra que la evaluación en línea puede ser implementada de manera segura y eficiente y puede resultar en un incremento en la motivación de los estudiantes. Gail (2003) añade que diversas investigaciones y reportes de docentes revelan que evaluar en internet puede resultar más fácil, rápido y amigable que hacerlo presencialmente con los métodos tradicionales, tanto para quien elabora la evaluación como para quienes son evaluados. Sin embargo, también es cierto que las actividades de evaluación sólo adquieren verdadero peso cuando están integradas a las experiencias de aprendizaje del curso (véase Woit y Mason, 2003).

La expansión o potenciación que han sufrido las actividades de evaluación del aprendizaje en la educación virtual, ocurre básicamente porque internet ha abierto la posibilidad de implementar procesos que antes eran imprácticos; un ejemplo claro de esto es que hoy el docente tiene la opción de aplicar los diferentes esquemas de evaluación y las diferentes fórmulas de calificación que se vieron anteriormente, sin necesidad de que domine sus bases matemáticas. Los sistemas informáticos cuentan también con otra poderosa capacidad: herramientas para ofrecer realimentación prácticamente en tiempo real durante las sesiones de evaluación. Además, particularizar la educación, adaptándola a las necesidades de cada individuo, nunca estuvo más cerca de ser una realidad como ahora.

Gail (2003, p.32-34) menciona otros cambios que ocurren en el ambiente de internet:

1. Los exámenes en línea son ahora más apropiados para la evaluación formativa.

2. Pueden aplicarse otros esquemas de puntaje: Por ejemplo, se ganan puntos por el sólo hecho de realizar una actividad, aunque ésta en sí no tenga calificación; o la calificación de un alumno puede incluir la coevaluación y la autoevaluación.
3. El contexto a distancia exige tomar en consideración la facilidad de copiar, por lo que la mejor práctica es aplicar exámenes a libro abierto manteniendo un control del tiempo.
4. Ciertas cuestiones técnicas son una completa novedad. La evaluación en línea, especialmente si las actividades están circunscritas a un tiempo restringido, exige tomar provisiones para resolver problemas técnicos, como pérdida de conexión o conexiones lentas.
5. Los sistemas facilitan la posibilidad de que las actividades se resuelvan tantas veces como se requiera, que cambie la secuencia de actividades e incluso que se generen ítemes aleatorios.
6. Internet permite asociar material multimedia a una actividad de evaluación para crear una experiencia multisensorial y multidireccional.
7. Las herramientas de comunicación facilitan la programación de actividades para ser resueltas en colaboración, como los webquest y los exámenes en equipo. Esta modalidad colaborativa ha recibido mucha atención últimamente y entre los hallazgos se menciona que cuando los estudiantes presentan exámenes en equipo al principio de un curso, al presentar exámenes individuales al final sus resultados tienden a mejorar.

Estándares de software

Definiciones y aportaciones

El segundo gran componente del marco teórico de esta investigación lo constituye el tema de los estándares, y lo primero que puede decirse de ellos es que están en todos lados: el metro es un estándar de medida y el dinero de intercambio, los idiomas son estándares de

comunicación y las luces de semáforo de seguridad. La estandarización produce un gran impacto en la vida, aun cuando se sabe muy poco acerca de los propios estándares o los procesos para crearlos (Breitenberg, 1987). Pero ni siquiera es raro que las personas formen parte inconsciente del movimiento que los hace surgir. Por ejemplo, ocurre en el mundo informático que las compañías frecuentemente lanzan una tecnología propietaria que no trabaja bien con otras y cuando además no satisface las necesidades de los usuarios finales, el mercado acaba impulsado a los líderes de negocios, de escuelas y de los gobiernos a trabajar juntos para desarrollar estándares comunes (Masie, 2002).

Así que los estándares surgen de la actividad de los organismos reconocidos que se dedican formalmente a producirlos (estándares *de jure*) o de la fuerza de la práctica (estándares *de facto*). Por ejemplo, el estándar de interconexión de sistemas abiertos (Open Systems Interconnection) de la ISO es un estándar de jure, mientras que el sistema operativo DOS de Microsoft, fue un estándar de facto (Rada y Craparo, 2000; ver Updegrove, 1995). El estado ideal, dice Masie (2002), es que el estándar de jure sea también el estándar de facto (por ejemplo, el http).

Ahora bien, un estándar es por naturaleza tanto descriptivo como prescriptivo. Es descriptivo o definitorio porque refleja las creencias y el modo de hacer de un área específica, y es prescriptivo porque pretende que estas creencias y procesos comunes sean adoptados por todos. Esta adopción ocurrirá, excepto en el caso de que se imponga por decreto, si el estándar realmente satisface las necesidades de quienes lo apliquen (Moore, 2000; Rehesaar, 1996).

Los estándares en la industria computacional siempre se han basado en el éxito en tres campos (Stone, 1995): Primero, el económico, ya que una industria que controle la mayor parte del mercado convertirá a sus productos en los estándares de facto; segundo, el campo de

las uniones industriales, que suelen ser de dos tipos, las que se basan en un nicho para promover un producto, o las horizontales, que impulsan alguna tecnología; y tercero, el campo de las acreditaciones oficiales, respaldadas por gobiernos u organizaciones reconocidas.

Definiciones

Para Moore (2000, p. 6) un estándar puede ser:

1. Un objeto o medida de comparación que define o representa la magnitud de la unidad;
2. Una caracterización que establece tolerancias o requerimientos permitidos por categorías de ítemes; o
3. Un grado o nivel de excelencia o cumplimiento requerido.

Breitenberg (1987) lo condensa en una sola definición: “Una prescripción de reglas, condiciones o requerimientos concernientes a la definición de términos, clasificación de componentes, especificaciones de materiales, desempeño u operaciones, delineación de procedimientos, o medida de cantidad y calidad de materiales, productos, sistemas, servicios y prácticas”.

Cualidades de un buen estándar

Para el usuario final el origen de un estándar o su caracterización es menos importante que la utilidad que le ofrece; incluso para algunos la calidad del software está subordinada al hecho de que existen estándares, cuya aplicación obligada por parte del productor, los ayudarán a usar y aprender más fácilmente dicho software. A pesar de este desinterés en los detalles, el mercado sigue prefiriendo un buen estándar a uno mediocre, así que la cuestión es ¿qué caracteriza un buen estándar? y, desde el punto vista de quienes los proponen, ¿qué debe

incluirse en el proceso de propuesta para tener mayor probabilidad de obtener un buen estándar?

Para Updegrave (1995, p. 143) un buen estándar debe al menos mostrar las siguientes propiedades:

1. Su calidad técnica debe ser alta; es decir, ser práctico al implementarse, eficiente en su operación y resultados, compatible con otros estándares y realidades técnicas, etc.
2. Debe ser una solución efectiva a los problemas que dieron lugar a su estudio y elaboración.
3. Debe ser oportuno, para que se produzca en un momento en el que aún resulte útil.
4. Debe ser adoptado ampliamente. Esta es la característica que, para Updegrave, es la más importante.

Necesidades satisfechas por los estándares

Los estándares pueden clasificarse de muchas maneras, ya que pueden verse desde una amplia variedad de puntos de vista: el dominio al que pertenecen, su finalidad, los beneficios que procuran, su origen, etc. La International Standardization for Organization (ISO) para fines prácticos, por ejemplo, considera ocho tipos basados en su propósito: estándares básicos, de terminología, de prueba, de productos, de procesos, de servicios, de interfases y de listas de requerimientos de datos para productos y servicios (Breitenberg, 1987). A pesar de esta variedad, se puede decir de ellos que prácticamente cualquier campo se beneficia al implementarlos, independientemente del modo, formato y alcance de las normas y prácticas que exprese. Los rasgos comunes que poseen los estándares permiten ilustrar el por qué de su surgimiento al hablar de un tipo en particular, que en el caso presente se trata de los estándares de la ingeniería del software.

Así como la libertad de expresar lo que se desea depende fundamentalmente del acuerdo en hablar los mismos idiomas, así los estándares son el fundamento lingüístico común sobre el que se construye la diversidad del mundo. Para que sean útiles para el software, los estándares deben estar disponibles a lo ancho del mundo y estar libres de restricciones que impidan su amplia adopción (Rosen, 2002).

Se esperaría que los estándares de la ingeniería del software contribuyan a la implementación de un modelo formal, fiable y efectivo para el desarrollo, operación y administración de los sistemas de software (Moore, 2000), ya que hacen las siguientes contribuciones:

1. Proveen una nomenclatura para conceptos complejos.
2. En ausencia de pruebas con reconocida validez científica, ofrecen criterios de medición y evaluación técnicas que son consideradas de valor para la sabiduría del consenso.
3. Registran el consenso de una comunidad respecto a las “mejores prácticas”, esto es, técnicas ampliamente consideradas como efectivas. Algunos estándares comparten técnicas que pueden conducir a mejoras cualitativas para desarrollar software mejor, más rápido o menos costosamente, por ejemplo.
4. Definen con claridad el campo de las responsabilidades de quienes participan en procesos estandarizados.
5. Proveen un marco de comunicación para los diferentes actores, reduciendo las malinterpretaciones y acortando el tiempo (y el texto) necesario para alcanzar acuerdos.
6. Son la base de la certificación. Muchas organizaciones están interesadas en mostrar explícitamente que sus prácticas institucionales se conforman con una constelación de las mejores prácticas y principios, y procuran obtener constancia de ello; estos certificados están

basados en estándares y suelen ser formulados por autoridades expertas y a veces independientemente certificadas.

En suma, concluye Moore, los estándares pueden mejorar los producto de software, proteger al comprador, proteger el negocio, incrementar la disciplina profesional, introducir tecnología y, como también dice Masie (2002), asegurar el intercambio y la interoperabilidad.

En busca de calidad

En general puede decirse que los estándares apuntan a obtener calidad. Es la percepción habitual asociar un estándar con la calidad de un producto. “Generalmente la gente piensa en la calidad como el hecho de conformarse y cumplir ciertas especificaciones continua y consistentemente” (Ward, 1999, p. 2). Por supuesto, agrega Ward, los usuarios pueden tener un punto de vista “trascendental”, como si la calidad de un producto fuera una “excelencia innata”, subjetiva e indescriptible; sin embargo, este punto de vista es rechazado por los ingenieros en general, y por los ingenieros de software en particular.

Por otro lado, aunque el desarrollo y uso de estándares tiene costos y puede implicar desventajas o riesgos, sus beneficios sobrepasan por mucho a las dificultades de aplicarlos o al abuso que podrían fomentar. Por ejemplo, como no es posible siempre evaluar individualmente los productos que se adquieren, la conformación de estos a estándares aceptados ofrece una fuente de información para calificar la idoneidad de un producto. “Los estándares están en el fondo de los métodos de producción en masa. Promueven una más efectiva y organizada interacción social, como es el caso de los colores de las luces del semáforo” y además, añade Breitenberg (1987), “los estándares permiten a la sociedad hacer un uso más efectivo de sus recursos y permiten una más efectiva comunicación entre todos los participantes de actividades, transacciones o procesos”. Incluso en el ámbito de la educación

virtual, los estándares apuntan a asegurar las cualidades deseables del software educativo (Tabla 5).

Finalmente, afirma Stone (1995), el propósito central en la ingeniería del software para un estándar es la interoperabilidad. Succi (1998) lo expresa de otro modo al decir que la “compatibilidad es la decisión estratégica clave en la producción del software”. Aunque a veces ser el pionero es el camino a seguir, hay también beneficios en actuar como los demás. Estos beneficios incluyen el intercambio de productos complementarios, facilidad de comunicación entre los usuarios de la misma red y ahorro en costos, especialmente debido a la producción en masa. Pero plegarse a lo que suelen hacer todos también inhibe el impulso por adoptar tecnologías innovadoras.

Tabla 5

Habilidades de los sistemas de elearning que los estándares ayudan a asegurar*

Interoperabilidad: La capacidad que tiene un sistema de trabajar en otro.

Reusabilidad: La facilidad de reutilizar los objetos de aprendizaje.

Gestión: La facilidad que brinda el sistema para tener la información concreta y correcta acerca del estudiante y de los contenidos.

Accesibilidad: La facilidad de que el estudiante pueda acceder fácilmente a los contenidos apropiados en el tiempo apropiado.

Durabilidad: La capacidad que tendrá la tecnología de eliminar la obsolescencia

* Castro Solís (2002); véase Masie (2002)

Historia y organizaciones

En el área de la ingeniería del software cierta fuente (ver Moore, 2000) identifica 315 estándares extensamente usados, que han sido creados y mantenidos por alrededor de 55 organismos internacionales, nacionales, sectoriales o profesionales (ver en la Tabla 6 la lista de los más reconocidos). Tan sólo la comisión de estándares para la ingeniería del software del Institute of Electrical and Electronics Engineers (IEEE) mantiene alrededor de 50, habiendo

invertido en cada uno entre 2 y 4 años, y entre 2 mil y 10 mil dólares por página. Alguien estima que en la ISO el proceso suele exceder los 7 años.

Uno de los primeros intentos conocidos de establecer formalmente un estándar en el mundo occidental fue el decreto de Enrique I de Inglaterra de 1120, por el cual la *ana*, o yarda

Tabla 6

Principales organismos productores o certificadores de estándares relacionados con la ingeniería del software

Estandarizadores generales

IEC: International Electrotechnical Committee

EIA: Electronic Industries Association

ISO: International Organization for Standardization

JCT1: Joint Technical Committee (comisión conjunta de la IEC y la ISO)

IEEE: Institute of Electrical and Electronics Engineers

IEEE Computer Society

ANSI: American National Standards Institute

AIAA: Institute of Aeronautics and Astronautics

NOM: Normas Oficiales Mexicanas establecidas por la Comisión Nacional de Normalización

Estandarizadores de la educación virtual

IEEE LTSC: IEEE Learning Technology Standards Committee

IMS: Instructional Management Systems Project

ARIADNE: Alliance of Remote Instructional Authoring and Distribution Networks for Europe

PROMETEUS: PROMoting Multimedia access to Education and Training in the EUROpean Society

AICC: Airline Industry CBT Comitee

SIIA: Software and Information Industry Association desarrollador del Schools Interoperability Framework (SIF)

ADL: Advanced Distributed Learning, desarrollador del Shareable Content Object Reference Model (SCORM)

CEN/ISSS: European Committee for Standardization/Information Society Standardization System

EDNA: Education Network Australia

DCMI: Dublin Core Meta-data Initiative

GEM: Gateway to Educational Materials

antigua, se convertía en la unidad de medida de su reino, debiendo equivaler a la longitud exacta de su brazo. Siglos más tarde, en las colonias inglesas en América, los funcionarios de la ciudad de Boston establecieron que se consideraría un delito la manufactura de ladrillos que no midieran 9 x 4 x 4 pulgadas. El que haya unidades de medida definidas ya era practicado en

el mundo antiguo; los arqueólogos han encontrado cilindros de piedra de alrededor del 7000 a.C. que eran usados para pesar en Egipto (Breitenberg, 1987).

Muchos campos del saber y de la industria han formalizado sus estándares desde hace siglos, especialmente a partir de la Revolución Industrial; pero aunque la informática es muy joven en comparación, las propuestas de normatividad para ella han tenido un crecimiento explosivo. Los primeros estándares específicos para la ingeniería del software se desarrollaron a principios de los 70 en oficinas del gobierno de Estados Unidos (Moore, 2000); y desde entonces se han sumado gran número de otras importantes organizaciones.

Principales organizaciones estandarizadoras

IEC. La Internacional Electrotechnical Commission fue establecida en 1906 con la misión de “promover, a través de sus miembros, cooperación internacional en todas las cuestiones de estandarización y asuntos relacionados, en los campos de la electricidad, la electrónica y tecnologías relacionadas” (Moore, 2000, p. 55). Forman parte de ellas representantes nacionales de los organismos de estandarización de los países correspondientes.

EIA. Fundada en 1924, la Electronic Industries Association (EIA) es una asociación comercial que representa a un amplio espectro de compañías en Estados Unidos relacionadas con la producción electrónica. Aunque la EIA está organizada en grupos respecto a productos electrónicos y líneas de mercado, sus actividades relacionadas con estándares son supervisadas por una organización única (Moore, 2000, p. 30).

ISO. Es el más conocido desarrollador internacional de estándares. La International Organization for Standardization es una federación no gubernamental de organismos de estándares nacionales, creada en 1947. Su dominio incluye actualmente la estandarización en

todos los campos excepto la ingeniería eléctrica y electrónica, cubierta por la IEC (ver Tripp, 1996, p. 149) y las telecomunicaciones, cubierta por la Internacional Telecommunications Union (ITU). La misión de la ISO es promover el desarrollo de la estandarización y actividades relacionadas en el mundo con vistas a facilitar el intercambio internacional de bienes y servicios, y para desarrollar cooperación en la esfera de la actividad intelectual, científica, tecnológica y económica. El trabajo de la ISO ha resultado en el desarrollo de 9,300 estándares.

Como en la IEC, los miembros de la ISO son organismos nacionales y su trabajo es desarrollado en una organización jerárquica de comisiones técnicas, subcomisiones y grupos de trabajo, el nivel más bajo en el cual los documentos son elaborados (Moore, 2000, p. 70-71). El trabajo de la ISO es llevado a cabo a través de unos 2,300 cuerpos técnicos en los que participan más de 20 mil expertos de todo el mundo (Breitenberg, 1987).

JCT1. Cuando la ISO y la IEC coincidieron en el área de las tecnologías de información (IT), dieron forma en 1987 a un grupo conjunto, el Joint Technical Committee, cuya misión era establecer la estandarización en el campo de las IT. Esta enorme tarea ha sido enfrentada dividiendo el JTC1 en 19 subcomisiones, los cuales a su vez se han dividido en grupos de trabajo (Rehesaar, 1996, p. 124; Tripp, 1996, 149). Una de esas subcomisiones, la séptima, conocida como SC7, o con su nombre largo ISO/IEC JTC1/SC7, es la mayor fuente de estándares internacionales para de la ingeniería del software (Moore, 2000, p. 82).

IEEE. El Institute of Electrical and Electronics Engineers es la sociedad técnica más grande del mundo con miembros en alrededor de 150 países; afirma que es la responsable de haber publicado el 25% de todos los documentos técnicos de su área. Está organizada en 37

sociedades técnicas, algunas de las cuales son activas en el desarrollo de estándares. Estos surgen de la promoción de la dirección general y del interés de patrocinadores, quienes definen un área que requiere un estándar; posteriormente grupos de trabajo dan forma al estándar buscado. La IEEE produce tres tipos de documentos: estándares, prácticas recomendadas (*best practice*) y guías, que se diferencian del grado de obligatoriedad de la normativa que proponen. Los estándares son obligatorios (generalmente se caracterizan por la palabra "debe"); las prácticas recomendadas presentan procedimientos o posiciones preferidas por la IEEE (generalmente tienen la palabra "debería"); y las guías sugieren prácticas alternativas (generalmente iniciadas con la palabra "podría").

La IEEE produce cualquiera de los tres documentos en versiones de prueba para dos años, luego de los cuales pueden ser aprobados para un uso permanente o devueltos a estudio. Ya aprobados tienen una vida de 5 años y después se retiran, se confirman sin cambio o se revisan (Moore, 2000).

IEEE Computer Society. Es la asociación computacional más grande del mundo; fundada hace 50 años, es ahora la sociedad técnica más grande de la IEEE. Sus proyectos de estándares son canalizados a comisiones y patrocinadores; la correspondiente a la ingeniería del software es la Software Engineering Standard Committee (SESC), que se remonta a 1976. La SESC mantiene actualmente un inventario de alrededor de 50 estándares clasificados en varias categorías (Moore, 2000).

ANSI. Dice Moore (2000, p. 25-26) que "lo más importante que debe saberse de la American National Standards Institute es que no crea estándares". Es una organización privada, sin fines de lucro que coordina el desarrollo voluntario de estándares nacionales de

Estados Unidos y sirve como el cuerpo nacional de estándares en Estados Unidos tal como la ISO y la IEC. La aprobación de un estándar por la ANSI implica que sólo se juzgó que el proceso de desarrollo fue aplicado apropiadamente, pero no emite ningún juicio técnico. La ANSI ni siquiera verifica que el estándar pertenezca al dominio de la organización que lo presenta o que sea consistente con otros estándares nacionales; tan sólo se apoya en la declaración de la organización presentadora.

AIAA. La American Institute of Aeronautics and Astronautics es la principal sociedad que sirve a la profesión aeroespacial. Aunque sus estándares están orientados hacia los requerimientos particulares del aeroespacio, su utilidad ha hecho que algunas veces sean aplicados más generalmente (Moore, 2000).

NOM. En México el responsable de establecer estándares y vigilar su implementación es la Comisión Nacional de Normalización, cuya base legal se encuentra en la Ley Federal sobre Metrología y Normalización, promulgada en 1992 (*Diario Oficial de la Nación*, 1 de julio de 1992). Entre otros objetivos tiene el de actuar en las áreas de normalización, certificación, acreditamiento y verificación (artículo 2). Esta ley contempla dos tipos generales de normas o estándares: Las Normas Mexicanas, de adopción voluntaria y las Normas Oficiales Mexicanas (NOM), de carácter obligatorio (artículo 3). Sus antecedentes se remontan a los años 40 cuando una dependencia de la Secretaría de Comercio coordinaba la normatividad enfocada a los servicios, los materiales y a los métodos de ensayo. El nombre de Normas Oficiales Mexicanas existía antes de la ley de 1992 y se utilizaba para productos de distintas industrias, con carácter voluntario, enfocado a la calidad (Pineda, 2002).

Estandarizadores de la educación virtual

Algunos organismos más, o extensiones de los mencionados anteriormente, se han especializado en el campo de la educación virtual para emitir los estándares correspondientes, muchos de los cuales se relacionan con la ingeniería del software. Stacey (febrero de 2003) anota que tales organismos a veces parecen más preocupados por la carrera de producción de contenidos en línea al crear sus especificaciones y estándares, enfocados en los metadatos, cuando en realidad el mercado está empujando hacia el lado humano de la educación. Como abunda este autor, no se trata sólo de lograr eficiencia técnica, sino alcanzar el blanco, a saber, la interacción en el aprendizaje.

En el desarrollo de estándares de la ingeniería del software para la educación virtual los primeros trabajos los realizaron estas organizaciones: ARIADNE en Europa, el Dublin Core, la IEEE, la AICC y el IMS. Estos grupos trabajaban simultáneamente pero sin coordinarse. Un intento de integración se ha llevado a cabo con SCORM, que es un Modelo de Referencia o conjunto de especificaciones y estándares de contenidos, tecnologías y estándares de elearning (Masie, 2002).

IEEE LTSC. La IEEE Learning Technology Standards Committee acredita los estándares de elearning para Estados Unidos; es decir, todos aquellos que provengan del Instructional Management Systems Project (IMS) y la Airline Industry Computer-based Training Comitee (AICC). Su propuesta de estándar está actualmente en revisión y se conoce como especificación Learning Technology Systems Architecture (LTSA). Esta propuesta establece un marco general bien definido para el análisis, diseño, implementación y evaluación de los entornos de enseñanza–aprendizaje. La arquitectura LTSA se compone de:

1. Usuario o alumno

2. Sistema de evaluación o seguimiento
3. Sistema tutor o profesor
4. Sistema de distribución de contenidos
5. Sistema de gestión y almacenamiento de datos en el que se recoge toda la información referente a la evolución del alumno; y
6. Biblioteca de contenidos educativos (Castro Solís, 2002).

Consortios. Algunos grupos, formales e informales, se asocian en esta forma. Una de las virtudes de los consorcios es que tienen el potencial para crear estándares rápidamente, debido a que normalmente los participantes están altamente motivados para lograr resultados (Updegrave, 1995). El caso más visible es el IMS.

IMS. El Instructional Management Systems Project es un consorcio mundial que ha desarrollado varias especificaciones para facilitar el aprendizaje en línea. Está conformado por varias compañías. Toman como base especialmente el trabajo de la AICC y su estudio se dirige al desarrollo de un conjunto de metadatos que puedan utilizarse en cualquier parte y que pueda definir un ambiente de elearning, incluyendo las características del usuario (Castro Solís, 2002; Learning Resource iNterchange frequently asked questions, 2004).

ARIADNE. El Alliance of Remote Instructional Authoring and Distribution Networks for Europe es una asociación de industrias centradas en los aspectos de estándares para el elearning europeo, que trabaja conjuntamente con PROMETEUS y la red europea de centros escolares (European Schoolnet, EUN). En el 2000 lanzó su versión oficial 3 de metadatos basada en la especificación de metadatos de objetos de aprendizaje o Learning Objects Metadata (LOM) (Castro Solís, 2002).

PROMETEUS. El nombre es el acrónimo de PROMoting Multimedia access to Education and Training in the EUropean Society; es una iniciativa de la Unión Europea, surgida en 1999, que pretende también incluir como especificaciones la multiculturalidad y el multilingüismo (Castro Solís, 2002). Se ha convertido en un foro de expertos de opinión donde los actores de un amplio rango de trasfondos profesionales, culturales y lingüísticos se reúnen para construir una masa crítica en el campo de la tecnología educativa y sus aplicaciones (Masie, 2002).

AICC. La Airline Industry CBT Comitee se formó con el propósito de crear estándares para la aviación y fue uno de los pioneros en tratar de proponer y crear estándares para el elearning, basándose en los entrenamientos en línea, pruebas y lecciones (Castro Solís, 2002).

SIIA - SIF. La Software and Information Industry Association es la responsable de desarrollar el Schools Interoperability Framework (SIF), un estándar para la interoperabilidad de los aspectos administrativos y contextuales de la educación elemental (Software and Information Industry Association, 2003).

ADL - SCORM. La Advanced Distributed Learning es una iniciativa del Departamento de Defensa de los Estados Unidos, a la que se han unido otras entidades del gobierno, empresas e instituciones educativas con el fin de establecer especificaciones para distribución de contenidos educativos entre sistemas a escala mundial. El resultado es conocido como Shareable Content Object Reference Model o SCORM (Advanced Distributed Learning overview, 2004; Shareable Content Object Reference Model, 2004).

CEN/ISSS. Es la European Committee for Standardization/Information Society Standardization System. Su misión es proveer un espectro integrado y comprehensivo de servicios y productos orientados a la estandarización, con el fin de contribuir al éxito de la sociedad de la información en Europa (Masie, 2002).

EdNA o Education Network Australia, es un servicio que apunta a apoyar y promover los beneficios de internet para el aprendizaje, la educación y la capacitación en Australia, pero cuyas experiencias han servido de base para desarrollos similares en otras partes del mundo (Masie, 2002).

DCMI. La Dublin Core Meta-data Initiative es un foro abierto que se ha comprometido en desarrollar estándares de metadatos interoperables que soporten un amplio espectro de propósitos y modelos de negocios.

GEM. La Gateway to Educational Materials es un consorcio promovido por el Departamento de Educación de Estados Unidos que busca proveer a los educadores con acceso rápido y fácil a miles de recursos educativos en sitios de internet (Masie, 2002).

Estándares del elearning

En el ámbito internacional se destacan unos cuantos estándares para la educación virtual relacionados con la ingeniería del software. Forman entre ellos un encadenamiento que es sintomático de la tendencia actual en la estandarización, a saber, la de construir unos estándares sobre otros.

EML. El Educational Modelling Language es una propuesta de notación desarrollada por la Open University of Netherlands. Vale la pena describirlo con algo de detalle por la influencia que ha tenido a través del IMS.

Un grupo de investigadores de esa universidad holandesa centró su estudio en el concepto de diseño de aprendizaje (learning design), como parte del diseño instruccional; lo definieron simplemente como el “modelado de unidades de estudio”. Tattersall et al (2003) afirman que “el uso de la palabra ‘aprendizaje’ ayuda a enfatizar la variedad de aproximaciones al aprendizaje por encima y más allá del ‘enseñar e impartir conocimiento’” (p. 2-3) que se asocia más generalmente con el diseño instruccional.

Su investigación los llevó a concluir que faltaba una forma estandarizada de describir el proceso de aprendizaje virtual, por lo que muchos diseñadores frecuentemente usan HTML o lenguajes propietarios para codificar secuencias de actividades que deben realizar los alumnos, para permitir que herramientas de comunicación sean incorporadas en esas situaciones, para almacenar el resultado de las interacciones, etc. Pero tal práctica tiene muchas dificultades: al documentar sus estrategias de enseñanza, al establecer y adherirse a procedimientos descritos para lograr consistencia en la documentación, al tratar de asegurarse de que la calidad de la enseñanza puede ser lograda dentro y fuera de las organizaciones y al reutilizar elementos de materiales de enseñanza existentes (Tattersall et al, 2003).

Al mismo tiempo, estos estudios consideraron las ideas existentes en torno a los objetos de aprendizaje, o paquetes de estudio autónomos y reutilizables, y descubrieron el creciente interés en estandarizar los metadatos y el empacamiento de los contenidos, en una clara tendencia a conceder la supremacía a la reutilización de contenidos a costa de relegar la relación persona-persona y persona-contenido (Stacey, febrero de 2003). Pero, como agrega

Stacey en otro sitio (enero de 2004), los repositorios de objetos de aprendizaje son una parte esencial de la arquitectura, pero sólo una parte.

Koper (2001), uno de los principales investigadores, se preguntó entonces si el modelo de objetos de aprendizaje y paquetes provee medios suficientes para construir unidades de estudio completas, flexibles y válidas para ser publicadas a través de sistemas de educación. Su respuesta fue rotundamente negativa. Desde una perspectiva educativa no es suficiente tener objetos de aprendizaje y metadatos como tales, “el principal problema es que los objetos de aprendizaje no son tipificados para su uso en el contexto de una unidad de estudio. Para ponerlo de otra manera: Hace falta un marco que los contenga”.

La solución propuesta fue desarrollar, a fines de los 90, un sistema de notación que fuera capaz de describir unidades de estudio de una manera neutral, pedagógicamente hablando; esta notación fue conocida como EML. Una vez descritos en EML, las unidades de estudio pueden ser interpretados (*played*) por un componente de software basado en EML (*player*), de la misma manera que el HTML es interpretado por un navegador (Tattersall y Koper, 2003). Se espera que el EML juegue el mismo papel que desempeña la notación musical, antes de la cual sólo podía preservarse la música cantando y tocando; un papel que la computadora sea capaz de interpretar (Koper, 2000).

De hecho, para que pueda ser leído el EML se ha formulado como un documento XML y está basado en la noción de componentes o bloques de construcción (*building blocks*), que son ensamblados en nuevas configuraciones para así producir nuevas unidades educativas. Los componentes son unidades separadas que pueden ser desarrolladas independientemente y ser reutilizadas en diferentes aplicaciones (Hermans, Koper, Loeffen, Mandelveld y Rusman, 2000).

Llama la atención, para los fines de esta investigación, que según la documentación del EML (Hermans, Koper, Loeffen, Mandelveld y Rusman, 2000) pueden modeladas preguntas con EML para usarlas en estructuras de preguntas frecuentes (FAQ) o para autoevaluaciones, pero exámenes formales completos no pueden ser representados, ya que el EML no modela el contexto administrativo o el manejo de un examen formal.

IMS. Este consorcio de industrias del software y organismos educativos ha desarrollado varias especificaciones para el elearning, entre ellos el correspondiente al diseño del aprendizaje (Learning design), derivado del EML. Es conocido como IMS LD y su versión 1.0 fue aprobada el 10 de febrero de 2003 (Tattersall y Koper, 2003). La razón para basarse en el EML es lograr especificaciones independientes del modelo pedagógico elegido. “El IMS LD provee un nivel de abstracción al proceso, ofreciendo constructos genéricos a diferentes acercamientos pedagógicos. Usando el lenguaje, los diseñadores pueden hablar en términos pedagógicos en lugar de tecnológicos” (Tattersall et al, 2003).

El IMS LD no se basa en los contenidos sino en las actividades de aprendizaje y el logro de objetivos de aprendizaje. De hecho, de acuerdo con el modelo del EML, admite que el aprendizaje puede ocurrir sin objetos de aprendizaje, porque el aprendizaje es diferente del consumo de contenido. La especificación también reconoce que el aprendizaje ocurre cuando los alumnos cooperan para resolver problemas en situaciones sociales y de trabajo (Tattersall y Koper, 2003).

El IMS ha elegido una estructura de estándares interrelacionados, de modo que el IMS LD debe acudir a otros para ser funcional, entre los cuales se encuentran: empaquetamiento de contenidos (*Content Packaging*), metadatos de los recursos de aprendizaje (*Learning Resource Meta-Data*), y otros (véase IMS Global Learning Consortium, 2003).

Es ilustrativa también la interrelación que guarda otra especificación del IMS, el IMS Learner Information Package (LIP), con estándares de otras organizaciones. El LIP se ha derivado en parte de las versiones 5 y 6 de la especificación IEEE Public and Private Information, que es el intento de la comisión P1484 de la IEEE (IEEE Learning Technology Standardización Committee P1484) de definir un alumno “portable”. También guarda relación con el SIF, aunque éste se enfoca más bien en permitir la interoperabilidad entre los sistemas administrativos de las escuelas y el LIP en la información educativa del alumno. Esta última información se estructura de una manera relacionada con el ANSI TS 130 Student Educational Record (IMS Learner Information Package Best Practice & Implementation Guide. Final Specification. Version 1.0).

SCORM. El Shareable Content Object Reference Model, cuyo desarrollo está encabezado por la ADL, no es realmente un estándar sino un modelo de referencia que sirve para probar la efectividad y aplicación en la vida real de una colección de especificaciones y estándares individuales. SCORM trabaja con organizaciones estandarizadoras, como AICC, IMSS y IEEE para integrar sus especificaciones en un modelo cohesivo y utilizable, y define las interrelaciones clave entre los estándares (Masie, 2002).

Este modelo se aplica para definir y usar objetos de aprendizaje, con el fin de compartirlos fácilmente entre diferentes sistemas computacionales de educación (*Learning Management Systems*, LMS). Fue el resultado de la preocupación del Departamento de Defensa de Estados Unidos por superar los problemas que enfrentaba en sus programas de educación a distancia. A través del ADL ha logrado proponer un modo de representar la estructura de un curso, una *application program interface* (API), el modelo de datos de un

LMS, una especificación para lanzamiento de contenidos y una especificación para registros de metadatos para todos los componentes de un sistema.

El propósito general del SCORM es ofrecer acceso a educación y capacitación de alta calidad, conformada a las necesidades individuales, a un costo accesible en cualquier parte y en cualquier momento.

Los requerimientos funcionales que se buscan son:

1. Ser accesible desde múltiples localizaciones remotas, mediante el uso de metadatos y estándares de empacamiento.
2. Ser adaptables, al acomodar la instrucción a las necesidades individuales y de la organización.
3. Ser costeable, al incrementar la eficiencia y productividad del aprendizaje a la vez que se reduce el tiempo y los costos.
4. Ser durable a pesar de las revisiones de sistemas operativos y software.
5. Ser interoperable a través de múltiples herramientas y plataformas
6. Ser reutilizable mediante el diseño, administración y distribución de herramientas y contenidos de aprendizaje a través de aplicaciones múltiples (Shareable Content Object Reference Model, 2004; Advanced Distributed Learning overview, 2004; Castro Solís, 2002).

LRN. Learning Resource iNterchange es una implementación de Microsoft de las especificaciones del Content Packaging 1.1 y Metadatos 1.2 del IMS, que es estándar de la industria para el intercambio de contenido de aprendizaje en línea (Learning Resource iNterchange frequently asked questions, 2004). Esta empresa es miembro del consorcio IMS y contribuye a sus estándares, pero, como otras empresas más, ha partido de alguna

especificación y la ha extendido para implementarla, heredando así conceptos de los que partieron en el consorcio, como las ideas del EML.

Establecimiento de estándares

Casi cada organización posee sus propios estándares internos (Ward, 1999), resultando en la práctica tan abundantes y variados como la forma de elaborarlos e implementarlos. Pero en el ámbito global, donde la necesidad de consenso y la fuerza del mercado son determinantes, se hace obligado ser más definido en cuanto a todo el proceso de estandarización; por ello, entre los organismos más reconocidos los dos métodos más comúnmente usados para lograr el consenso en el desarrollo de estándares son las comisiones y las campañas (Breitenberg, 1987):

Comisiones. Los estándares se sujetan a una revisión amplia y a la consideración de todos los interesados. Los requisitos de aprobación dependen de cada organización; por ejemplo, en algunas el consenso se define como el acuerdo de al menos 51% de los votantes; en otras se añaden requisitos a cumplir durante el proceso, un máximo de votos negativos, etc. Por este método se pueden lograr acuerdos que cuentan con una amplia aceptación, si bien a costa de invertir mucho tiempo.

Campañas. Esta forma es normalmente usada por organizaciones que han preparado estándares bajo sus procedimientos internos y que desean dotarlos con reconocimiento amplio; para ello los someten a votación ante un conjunto de organizaciones que representan una gama significativa de intereses, como productores, consumidores, entidades gubernamentales y otras. Las objeciones y propuestas de cambio pueden añadirse a una lista de campaña para resolverlas satisfactoriamente, aunque el riesgo es que surjan dos problemas: que la respuesta

sea escasa y que algunos de la lista de campaña, especialmente los consumidores, encuentren difícil participar debido a que no intervinieron en el desarrollo inicial, por lo que pueden no entender las razones o las implicaciones de ciertas provisiones.

Aunque no hay un proceso único para la creación de estándares de jure, suele aplicarse una secuencia altamente iterativa con cuatro fases típicas, presentadas en la Tabla 7 (Masie, 2002). Como se notará más adelante, estas cuatro fases están claramente representadas en los trabajos de la JTC1.

Tabla 7
Fases típicas del proceso de producción iterativa de estándares

fase	descripción	resultado
1. Investigación	Identificación y desarrollo de soluciones posibles a un problema dado	Propuesta de solución
2. Especificaciones	Desarrollo y documentación de la solución tentativa	Documentación de las especificaciones para ser distribuida y comentada
3. Pruebas	Pruebas y prototipos de las especificaciones	Documentación de las pruebas y propuesta formal
4. Estandarización	Consideración de la propuesta formal	De ser aprobada la propuesta, se publica en forma de estándar y se describe el proceso de acreditación

Las cuatro fases de la tabla 7 son paralelas a las que sigue el World Wide Web Consortium (W3C): Borrador de trabajo, recomendación candidata, recomendación propuesta y recomendación (Griffin 2002). Las fases del proceso del IMS son más numerosas, pero incluyen básicamente las mismas acciones que las cuatro fases (véase IMS Learner Information Package Best Practice & Implementation Guide. Final Specification. Version 1.0).

Siendo este organismo el generador por excelencia de estándares para la ingeniería del software, conviene describir el proceso que sigue en sus trabajos (véase Rehesaar, 1996, p. 124):

Los miembros de JTC1 son países, no individuos o empresas, de alguno de tres tipos:

1. Miembros participantes con capacidad de votar, conocidos como miembros P.
2. Miembros observadores o miembros O, que no tienen voto pero pueden asistir a todas las reuniones, hacer contribuciones y recibir documentos.
3. Miembros de enlace, que pueden ser internos o externos; entre los primeros están otros grupos ISO. No tienen capacidad de voto, pero pueden asistir a ciertas reuniones y recibir ciertos documentos, dependiendo de la categoría del enlace.

Esta comisión, y la ISO en general, produce dos grupos de documentos finales: los estándares internacionales (*international standard*, IS), que se revisan cada cinco años para conservarlos sin cambio, para incorporar revisiones que reflejen el estado actual de la tecnología o para desecharlos sin remplazarlos; y los reportes técnicos (*technical reports*, TR), de los cuales hay tres tipos:

TR1, que son borradores fallidos de comisiones o borradores de IS acerca de los cuales no pudo alcanzarse un consenso.

TR2, son técnicamente inmaduros y están aún bajo desarrollo.

TR3, que tratan de asuntos que no conviene publicar como IS, pero que son de cierto interés.

La JTC1 sigue un proceso de seis fases en el desarrollo de un estándar internacional:

Fase de preparación, 0, es una etapa preliminar opcional que se supera cuando el proponente presenta su iniciativa.

Fase de propuesta, 1; aquí las iniciativas se aceptan por mayoría simple en combinación con el compromiso de al menos cinco miembros P de participar activamente. Esta etapa puede durar alrededor de nueve meses.

Fase preparatoria, 2, durante la cual se desarrolla un borrador de trabajo, tomando al menos un año.

Fase de comisión, 3, en la que pasa una de las siguientes situaciones: El borrador en estudio es distribuido para ser discutido en reuniones de subcomisión, o es hecho circular ampliamente para ser comentado, o se presenta para su votación. Para que entonces avance se requiere un apoyo significativo; si no lo consigue y es rechazado, puede presentarse un borrador modificado en la siguiente plenaria o el JTC1 puede solicitar que se publique como TR1. Este proceso toma al menos un año y no debe pasar de cuatro.

Fase de aprobación, 4. El documento requiere de una mayoría de votos de dos tercios para ser aceptado, y que los votos en contra no representen más del 25%.

Fase de publicación, 5. El Grupo de Trabajo de Tecnologías de la Información (Information Technology Task Force, ITTF) de la JTC1 publica el documento como un estándar internacional.

La codificación

¿Qué ocurre con los estándares al ser formulados? ¿cómo se publican? A diferencia de lo que ocurre en el resto del mundo de la estandarización, en la ingeniería del software el proceso, los formatos y el modo de publicar sus estándares siguen en desarrollo y siguen siendo objeto de mucha discusión.

La codificación de los estándares de la ingeniería del software se enfrenta a desafíos particulares. En primer lugar, el sujeto de los estándares, el software, es inherentemente intangible y no está constreñido por las leyes comunes de la física. Segundo, la disciplina es relativamente nueva, comparada con otras disciplinas de la ingeniería, y muchos de sus más importantes conceptos aún son inmaduros. Tercero, no hay aún un cuerpo de conocimiento comúnmente aceptado que pueda servir de fundamento, ni ninguna entidad que se esfuerce en desarrollarlo. Finalmente, a diferencia de los estándares de las interfases de los productos, hay pocas fuerzas de mercado que causen convergencia en las tecnologías seleccionadas. (Moore, 2000).

Adicionalmente puede añadirse como una dificultad la abundancia y variedad de estándares. Precisamente en un comunicado de 1999 (ver Schmidt, 2000) la IEEE reconoció abiertamente que el crecimiento de su colección de estándares había llegado al punto de que era difícil percibir a simple vista las relaciones que hay entre ellos. Para resolver este problema creó una estructura de seis capas (Figura 2; véase también Moore, 2000), que presenta las consideraciones generales arriba y luego asuntos específicos y detallados abajo. Las capas superiores la conforman estándares que proveen terminología esencial, vocabulario y conceptos unificados usados por otros estándares de la colección. Las capas intermedias presentan estándares particulares y la última estándares de técnicas y prácticas específicas.

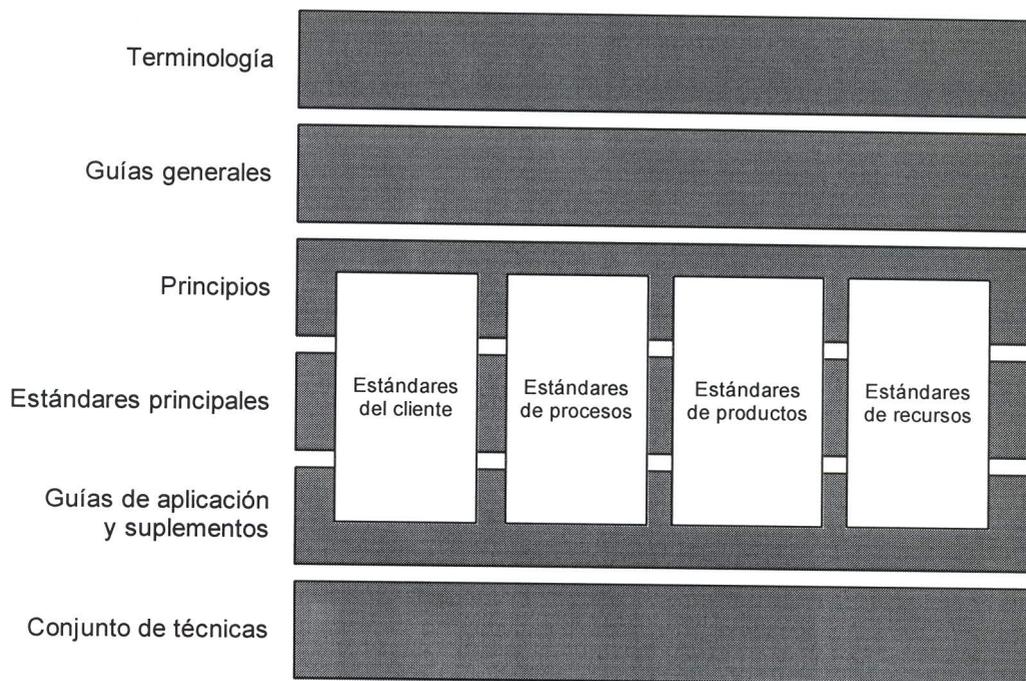


Figura 2. Organización estructural de los estándares de la IEEE

Además de esquematizarlos para ordenarlos y ver las relaciones entre ellos, debe resolverse el problema de la codificación. En ciertos aspectos son de utilidad las conocidas listas de revisión o checklist, pero es fácilmente evidente que se requiere un formato que exprese la complejidad del software.

Algunas de las vías que se han elegido, pudiéndose combinar entre ellas, son las siguientes:

Documentación. Este es un concepto que no se limita a los estándares, ya que es un resultado esperado en cada una de las etapas del ciclo de vida del software; por tanto, parece natural que los estándares se expresen en la forma en que se documenta el software habitualmente, para lo cual ya existen estándares (Hartmann, Huang y Tilley, 2001), como el

que define el Manufacture Supplier Relationship (MSR), que es un consorcio de varias compañías automotrices alemanas que con este estándar de documentación pretenden apoyar el desarrollo cooperativo entre las constructoras de autos y sus proveedores de sistemas electrónicos.

Los ingenieros de software ven la documentación como una ayuda para comprender la naturaleza funcional, el diseño de alto nivel y los detalles de implementación de aplicaciones complejas. Sin tal documentación habría que depender sólo del código, lo que puede ser una fuente de gran consumo de tiempo (Hartmann, Huang y Tilley, 2001).

Una de las motivaciones para especificar los requerimientos formalmente es que algunas notaciones hacen la revisión, diseño, implementación y desarrollo de casos de prueba más fáciles y más exactos. Además, la documentación formal ha mostrado que mejora la calidad del producto final. Otra motivación es que la notación formal permite un análisis formal para investigar si ciertas propiedades de seguridad se han preservado (Lutz, 2002).

La documentación puede presentarse en texto plano, en combinación con diagramas de flujo y control, o bien en UML o XML, como se ve en el fragmento del SIF en la Tabla 8.

UML. El lenguaje de modelado unificado (Unified Modeling Language) es un lenguaje estándar de diseño de software emergente. Provee un conjunto grande, útil y extensible de constructos predefinidos, está definido semiformalmente, tiene el potencial de soporte de herramientas, está basado en la experiencia con métodos de desarrollo orientados a objetos. UML es un lenguaje de modelado con una semántica y una sintaxis semiformal (Medvidovic, Rosenblum, Redmiles y Robbins, 1999). Ha sido adoptado como estándar por el Object Management Group, que es un consorcio internacional que integra a los principales

representantes de la industria de la tecnología de información orientada a objetos (Zavala, 2002).

Tabla 8

Fragmento de las especificaciones del SIF*

3.6.3 HTTP Request Headers

The following HTTP request and common headers defined in [RFC 2616] MUST be present in all SIF HTTPS messages sent by a client:

HEADER	DESCRIPTION	REQUIRED CONTENTS
Content-Length	The exact size of the attached payload (XML message)	
Content-Type	Describes the contents of the request. Firewall and web server programs can filter messages going through a network by examining this header.	application/xml;charset="utf-8"
Host	Specifies the Internet host and port number of the destination server	

Table 18: HTTP Request Headers

In addition to the headers above, a client may include a "Connection: close" header in the HTTP request if it wishes to close the current connection after receiving the response. If this header is included, the client must not send additional requests on this connection. The client should close the connection after receiving the response. The client may include additional headers; however, the server must be able to successfully process POST requests that only contain the required headers.

POST /MyPath HTTP/1.1

Content-Length: 420

Content-Type: application/xml;charset="utf-8"

Host: sifinfo.org:8000

<SIF_Message xmlns="http://www.sifinfo.org/infrastructure/1.x">

<SIF_SystemControl>

<SIF_Header>

<SIF_MsgId>56409F0C01FBD1C44300B4518E100765</SIF_MsgId>

<SIF_Date>20010411</SIF_Date>

<SIF_Time Zone="UTC-05:00">18:18:13</SIF_Time>

<SIF_SourceId>SifInfo_TestAgent</SIF_SourceId>

</SIF_Header>

<SIF_SystemControlData>

<SIF_Ping/>

</SIF_SystemControlData>

</SIF_SystemControl>

</SIF_Message>

*Software and Information Industry Association, 2003, p. 35

XML. El lenguaje de marcado extensible (eXtensible Markup Language) es un subconjunto del lenguaje estándar de marcado generalizado (Standard Generalize Markup Language, SGML), el cual es un estándar ISO (el 8879, para ser más preciso, Griffin, 2002) para definir y usar formato de documentos. La recomendación XML 1.0 fue desarrollada por el World Wide Web Consortium y publicada en febrero de 1998, considerándose el primer paso hacia la siguiente generación de la web, ya que permite a cada comunidad diseñar lenguajes que satisfagan sus necesidades particulares al mismo tiempo que ganan en interoperabilidad con la infraestructura general basada en XML. Además de otros desarrollos de ese mismo consorcio, uno de los más recientes y de mayor significado fue la publicación, en mayo de 2001, de las estructuras y los tipos de datos de los esquemas (World Wide Web Consortium).

XML se usa para documentos que contienen información estructurada; se procura así capturar tanto el contenido como el significado. Tres características lo han hecho más útil para esto que el más famoso lenguaje de marcado, el HTML: su extensibilidad, su estructura y su capacidad de validación (Hartmann, Huang y Tilley, 2001).

Los componentes de un documento XML son conocidos como elementos, atributos y entidades. Los elementos son constructos lógicos que definen partes de un documento. Un documento XML puede ser interpretado como una jerarquía de elementos, cada uno de los cuales representa una porción del documento. Los elementos tienen un tipo, pueden tener atributos y pueden tener contenido; este puede consistir de datos o de otros elementos anidados, o una mezcla de ambos. Por su parte, los atributos son los nombres de valores asociados con un elemento y las entidades son porciones de datos que conforman un documento XML (Hermans, Koper, Loeffen, Mandelveld y Rusman, 2000).

Una cualidad requerida de los archivos XML es la de estar bien formados, lo cual significa que siguen la sintaxis de la especificación XML 1.0; estas especificaciones son únicas y todos los archivos XML deben cumplirlas. Otra cualidad, aunque opcional, es ser válidos, lo que quiere decir que un archivo XML cumple con requerimientos particulares para cierto grupo de archivos XML. Estos requerimientos cumplen la función de ser una plantilla que expresa cuáles elementos y atributos puede o debe contener un archivo XML, qué tipo de datos representa cada elemento y cada atributo, en qué cantidad deben estar presentes, qué grado de obligatoriedad tienen, qué rangos de valores pueden tomar y cómo se relacionan estructuralmente los elementos y los atributos entre ellos (Holzner, 2001).

La verificación de la validez estructural se lleva a cabo con el fin de que las aplicaciones que lo consumen aseguren la consistencia de los datos contenidos en un documento XML.

Los primeros archivos de requerimientos fueron los Document Type Definitions (DTD). Sin embargo, pronto fueron evidentes sus limitaciones: No están escritos en sintaxis XML, no son compatibles directamente con los espacios de nombres (*namespace*) y ofrecen tipos de datos muy limitados (Griffin, 2002). La solución ha sido implementar los esquemas (XSD) porque están expresados en XML, se describen a sí mismos, pueden utilizarse directamente en internet, están optimizados para la interoperabilidad, son suficientemente simples para implementarse en un diseño sencillo, se coordinan con otras especificaciones del W3C, ofrecen muchos más tipos de datos simples y la opción para crear datos complejos (Griffin, 2002). Quizás sus únicas desventajas son que para algunos parecen ser más complejos que los DTD y todavía no son totalmente soportados por todos los navegadores.

En suma, la ventaja del XML es que se trata de un lenguaje ampliamente usado por todo tipo de aplicaciones y especialmente es aceptado por los navegadores de internet más comunes. Y además, los documentos XML pueden ser almacenados, consultados y actualizados en bases de datos relacionales, bien sea en un formato nativo o transformado a una representación interna de la base de datos (Hartmann, Huang y Tilley, 2001; Hermans, Koper, Loeffen, Mandelveld y Rusman, 2000).

La búsqueda de interoperabilidad y portabilidad de los sistemas está impulsando a la tecnología de la información, corriendo normalmente en internet, a inclinarse por el XML. Muchos estándares han seguido la misma ruta, especialmente los del campo de la educación virtual. Los casos son numerosos, pero destacan IMS y SCORM.

El proceso para utilizar XML es simple; Griffin (2002) lo resume de la siguiente manera (Figura 3):

1. Los datos se llevan al motor de XML y se genera un archivo XML
2. Puede aplicarse un DTD o un XSD para verificación
3. Puede aplicarse un XSLT para modificar la estructura del documento, con el fin de darle una nueva presentación o para adaptarlo a las exigencias que tienen distintos negocios.
4. Se genera un nuevo documento XML, transformado.

Los XSLT son hojas de estilo para XML, lo que son las CSS para el HTML. Son útiles básicamente para definir el modo en que se presentan gráficamente los datos del XML, y también para realizar cierto tipo de procesamiento de dichos datos, de modo que se adapten a las necesidades del usuario. Por ejemplo, un documento XML puede contener numerosos datos sobre empleados de una compañía y si sólo se requiriera desplegar en pantalla el nombre y su correo electrónico, un XSLT puede construirse para sólo utilizar y presentar tales datos.

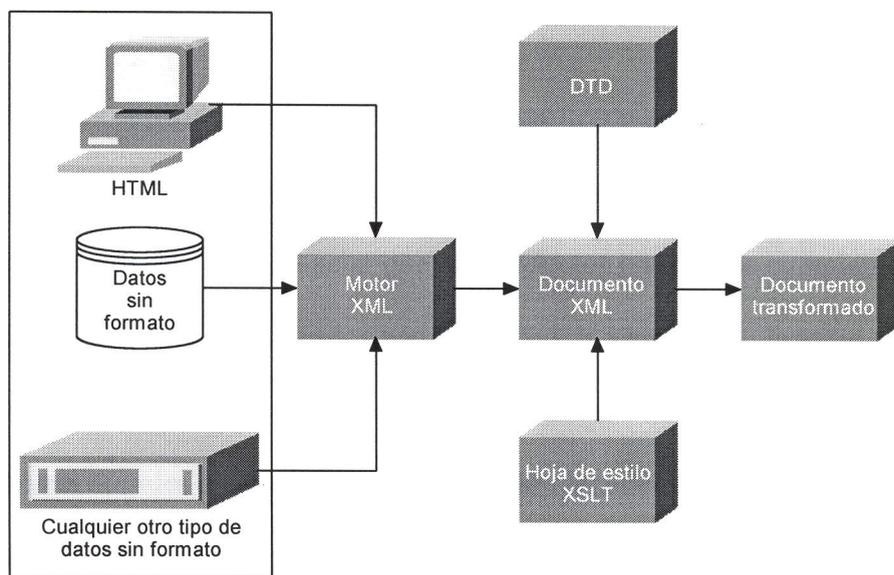


Figura 3. Proceso para utilizar el XML

Otros elementos. Frecuentemente los estándares en su enunciación presuponen la existencia, manejo o comprensión de otros estándares y adoptan elementos de expresión de ellos, como su notación. La ingeniería del software no es la excepción, así que es común encontrar casos como la introducción al SIF, que asume que para usar el estándar se tiene un manejo o buena comprensión de los protocolos de internet: TCP/IP, TLS/SSL y HTTP, del encriptamiento de datos y las firmas digitales basadas en certificados, de las bases de datos relacionales, de XML y sus herramientas de desarrollo (Software and Information Industry Association, 2003).

La certificación

La certificación de productos pretende confirmar que un producto en particular se conforma a uno o más estándares, de modo que se provea al consumidor con información

explícita o implícita acerca de las características y/o desempeño del producto (Breitenberg, 1987).

En México las certificaciones cobraron otro significado a raíz de la suscripción de tratados internacionales en la última parte del siglo XX, especialmente los de libre comercio. Esto dio lugar a la necesidad de evidenciar que los productos o servicios cumplen con un mínimo de estándares, ya sea relacionados con la calidad, la seguridad o la salud. Comienza a acostumbrarse entonces que estas evidencias sean obtenidas mediante certificados emitidos por instancias, personas morales o de tercera parte, también reconocidos en un marco jurídico (Pineda, 2002).

La certificación por terceros se aplica al proceso por el cual una organización, independiente del productor y del proveedor, evalúa la conformación del producto a uno o más estándares. Algunos de los métodos que usa son: prueba de tipo o inspección inicial, en la que los productos particulares no son probados sino sólo el plan de producción; prueba de auditoría seleccionando al azar muestras del mercado; supervisión del proceso de producción; investigación de campo, sometiendo a los productos a las condiciones de uso normal para descubrir fallas y sugerir acciones correctivas; prueba de lotes; y prueba total (Breitenberg, 1987).

Pronto fue evidente en el mundo informático, que trasladar ese proceso de certificación, que estaba volviéndose tan común en todas las áreas, al desarrollo de software, no era tan sencillo o automático. Sucede que la certificación del hardware está orientada primariamente al producto, mientras que la certificación del software está más orientada al proceso. Esto se debe a que es difícil realizar mediciones significativas del software en sí (Tripp, 1996).

Adicionalmente, debe recordarse que el software no “falla” en el sentido tradicional de la palabra. Si no lleva a cabo la tarea que se pretende, entonces tiene un error de diseño. Igualmente el software no se gasta ni se desajusta. Un error en el software, ya sea en su especificación o en el propio código, puede conducir a una pérdida de función o aun mal funcionamiento que contribuya a una condición de fallo del sistema. La probabilidad de una falla funcional depende de las características de desempeño de la plataforma más que de cualquier otro atributo de confiabilidad del software. Por esta razón, los cálculos de confiabilidad no son aplicados al software como se aplican al hardware, y no son un factor en el proceso de evaluación de la integridad del software (Tripp, 1996).

A pesar de las dificultades, la certificación del software involucra una evaluación posible respecto a ciertos criterios (Lutz, 2002). Desafortunadamente la ingeniería del software no ha madurado lo suficiente como para formalizar procesos de certificación ampliamente aceptados (Moore, 2000). Un modo de hacerlo sería verificando simplemente que el proceso siguió los cánones establecidos y que el producto se conforma a ciertos estándares de desempeño, calidad, seguridad, etc., pero no se ha definido con claridad cómo debe medirse esta conformación para cada caso.

Estado actual de la estandarización en la ingeniería del software

Mientras los estándares son una parte aceptada en las disciplinas maduras, como las ingenierías físicas y muchos de las áreas de negocios, no se han establecido del todo en el área de desarrollo del software. Su aceptación no suele ser automática, lo cual frecuentemente se debe al desconocimiento de cómo son desarrollados o a que se percibe su producción como demasiado lenta y burocratizada. Incluso algunos críticos dudan de la habilidad del estándar para representar realmente las necesidades de la comunidad (Rehesaar, 1996).

En este contexto, resulta revelador lo ocurrido en una universidad de los Estados Unidos (ver WebCT vs. Blackboard, 2002). La institución creó una comisión para evaluar dos aplicaciones para educación virtual; el estudio resultante describe detalladamente las propiedades de cada uno y lo que consideraban que podía lograrse con ellos en su institución educativa. Pero es notable que, aunque en varias partes se puntualiza que, especialmente WebCT, satisface estándares de IMS, lo que permite importar y exportar contenidos y datos, no se mencionan estos hallazgos como factor de decisión. El grupo finalmente se inclinó por WebCT debido a dos factores principales: (a) varios de los evaluadores ya lo conocen, se han acostumbrado a él y no encuentran en Blackboard un motivo de peso para cambiar; (b) las habilidades de WebCT sobrepasan a las de Blackboard. Los estándares fueron buscados y documentados concientemente, pero resultaron irrelevantes el tomar una decisión.

No todos los observadores concuerdan en que los estándares de la ingeniería del software han sido exitosos; “los estándares carecen de criterios de evaluación objetivos, involucran más al proceso que al producto, y no siempre están basados en resultados rigurosamente experimentales” (Moore, 2000, p. 9).

Por otro lado, aun cuando los estándares internacionales pueden ser el producto de un amplio consenso, suelen implicar importantes compromisos y frecuentemente evitan asuntos especialmente espinosos. Por ejemplo, dice Koper (2000), las comisiones que están ocupadas en estándares relacionados con el aprendizaje no suelen ocuparse de los modelos educativos; antes prefieren evitar complicaciones y abordar el asunto sólo técnicamente.

La complejidad es otra barrera. Al usuario de hoy de la ingeniería del software se le presenta un desconcertante conjunto de opciones, más de 300 estándares desarrollados y mantenidos por más de 50 diferentes organizaciones, con diferentes puntos de vista y

contextos. Esto a veces resulta en estándares inconsistentes, que se superponen y ocasionalmente se contradicen (Moore). Agrega Moore que libre de cualquier fuerza integradora de principios o productos dominantes, la mayoría de los estándares de la ingeniería del software son registros ad hoc de prácticas individuales que aseguran ser “la mejor”. Esto no es malo cuando cada estándar es considerado aisladamente, pero cuando se trata de verlos como un cuerpo, se encuentra que están sin integrarse, caprichosamente diferentes en detalle, sobrepuestos y ocasionalmente contradictorios.

La cultura puede representar un problema más. Muchos países desarrollan estándares locales para la ingeniería del software, que reflejan componentes culturales muy fuertes que impiden su uso en un contexto internacional (Abran, 1996).

La estandarización tiene el desafío de lograr un equilibrio interno pues aunque los estándares promueven la compatibilidad, también tienen la tendencia a estancar el progreso tecnológico y limitar la creatividad al resolver problemas. A eso se añade que las ruedas del proceso de estandarización internacional no se mueven muy rápidamente, y en el mundo del software y el hardware esto frecuentemente conduce a estándares inútiles e ignorados. Cinco o siete años para implementar un estándar para lenguajes de programación, por ejemplo, es un tiempo excesivo en el mercado de software donde las compañías liberan nuevos compiladores anualmente o incluso en plazos menores (Stone, 1995).

Muchas instituciones educativas hacen uso de estándares de software y hardware. Aunque los costos de soporte seguramente se incrementarían si no hubiera tales estándares, es difícil demostrar que los estándares actuales reduzcan significativamente la carga que representa el soporte, quizás porque los usuarios sienten las restricciones de los estándares, pero ven pocos beneficios obvios e inmediatos.

Ante este conjunto de dificultades o francas amenazas, la respuesta típica es minimizar la importancia de los estándares, pero la estandarización, tomada como estrategia sola, es la herramienta más poderosa para ofrecer un ambiente de información de alta calidad. Sin embargo, para explotar este potencial, es necesario establecer dos conceptos críticos: El primero es que ningún estándar puede satisfacer todas las necesidades, aunque puede establecerse un nivel mínimo de satisfacción de 80%, haciendo provisión para lidiar con el restante 20%. El segundo concepto es que los estándares deben abarcar el producto completo, no sólo el hardware y el software (Smith, 1997).

Tendencias y proyectos

La ausencia de un fundamento firme, empírico y científico para los estándares de la ingeniería del software ha sido una vulnerabilidad continua, una que no ha pasado desapercibida por sus críticos. Con la continua maduración del campo, se realizan intentos para tratar este problema. Entre los elementos que se invocan para superar los desafíos enunciados anteriormente está la adopción de un vocabulario como marco de trabajo, relaciones clave y otras restricciones a las que cada estándar debe adherirse, todo como parte de un requerimiento más básico, a saber, la identificación de un conjunto de principios fundamentales que sirvan para explicar y motivar las provisiones de varios estándares (Moore, 2000). En ese contexto pueden ubicarse algunas tendencias en el futuro de la estandarización de la ingeniería del software, particularmente evidentes en las colecciones de las organizaciones mayores involucradas en el desarrollo de dichos estándares. Destacan las siguientes:

1. Enfocarse en desarrollar principios fundamentales para la ingeniería del software.

2. Esfuerzos cooperativos de estandarización con disciplinas relacionadas pero más maduras tales como ingeniería de sistemas y administración de proyectos. En otras palabras, aprovechar lo hecho en campos relacionados.

3. Las llamadas pruebas SPICE, o Software Process Improvement and dEtermination project, que son esfuerzos para validar empíricamente los mecanismos de evaluación del proceso.

4. En años recientes, sin embargo, se ha visto una tendencia hacia el reconocimiento de estándares clave que proveen un marco que puede ser elaborado por otros más. Los ejemplos incluyen el ampliamente reconocido marco de trabajo de la administración de la calidad de los estándares ISO 9000 y el marco de referencia del ciclo de vida del proceso del estándar ISO/IEC 12207 acerca del ciclo de vida de los procesos.

5. Otra tendencia ha sido la de construir algunas colecciones sobre las contribuciones de otros (Moore, 2000, p. 16-17). Como se vio anteriormente, el EML es la base del IMS LD, y este a su vez ha incluido en el SCORM.

También vale la pena, como ilustración, mencionar algunas propuestas en campos específicos, que demuestran no sólo el interés y la aceptación de la estandarización, sino la diversidad de los campos de desarrollo:

Reciprocidad entre estándares y educación. En muchas disciplinas de la ingeniería, los contenidos de enseñanza se basan en un cuerpo de conocimiento acordado por un conjunto de certificadores, como los colegios de ingenieros, así que es razonable incorporar al plan de estudio la revisión formal y completa de los estándares de la ingeniería del software (Abran, 1996).

Estándares de información. Los estándares tradicionales se enfocan en los datos, como símbolos o hechos fuera de contexto, sin un significado directo e inmediato, por lo que las herramientas derivadas pueden manipular caracteres pero no interesarse en si representan poesía griega o ecuaciones químicas. Esto último es conocimiento, o información estructurada con un significado. Una veta de desarrollo sería estándares orientados hacia la entidad intermedia entre los datos y el conocimiento, la información, que son datos vistos en un determinado contexto interpretativo (Smith, 1997).

Estándares integrales. Es claro que los nuevos estándares no pueden ser desarrollados sólo por una organización tecnológica o de apoyo. Una solución integral requiere trabajo de equipo y el patrocinio de todos los grupos relacionados con el estándar. Por tanto, el desarrollo debe llevarse más allá del dominio tecnológico e involucrar aspectos no tecnológicos (Smith, 1997). Los estándares para el elearning, como suma del esfuerzo pedagógico, informático, escolar, comercial y gubernamental, pueden ser un beneficiario de esta propuesta integradora.

CAPÍTULO III

METODOLOGÍA

La presente tesis corresponde a lo que se conoce como proyecto en la Universidad de Morelia, si bien en sentido estricto las tesis sólo son de un tipo (véase Anderson, Durston y Poole, 1970; Turabian, 1996; y Zorrilla y Torres, 1986). Se concibe como un proyecto porque pretende implementar de manera concreta la solución a un problema, basado en metodologías, técnicas y herramientas predefinidas. La solución buscada es un estándar de la ingeniería de software que, como se ha visto, es una norma a la que se tienen que conformar el software, su proceso de desarrollo o ambos.

Para formular un estándar se puede seguir una gran variedad de caminos, ya que casi cada organización tiene sus propios procedimientos, que tienen significado sólo en su contexto. Sin embargo, Moore (2002) destaca como una tendencia muy notoria la práctica de basar un estándar en otros preexistentes, procedan o no de la misma organización. Esta investigación sigue esa práctica, aunque en pequeña escala, combinando elementos de dos fases del proceso de producción de software: el análisis de requerimientos y el diseño arquitectónico, fases reconocidas en el ciclo de vida del software (Davis, 1983; *Proceso del ciclo de vida del software según la ISO*), con el proceso de creación de estándares que se describió en el capítulo anterior (véase la sección “Establecimiento de estándares”).

En realidad puede afirmarse que las prácticas de los estandarizadores internacionales han llegado a conformar una suerte de *metaestándar* (esto es, una especificación de

estándares) de tres componentes: (1) El proceso de desarrollo; (2) el uso de XML para codificar y otros lenguajes de modelado, como UML, para construir los modelos; y (3) las expectativas respecto a un estándar.

La metodología o proceso se discute en este capítulo; el código y los modelos se tratan en el capítulo siguiente; y las expectativas, expresadas en términos de satisfacción de necesidades y de calidad esperada, están incorporadas en el rigor seguido y en la fundamentación del proceso y su producto que ha estado enunciándose desde el capítulo II.

Proceso de desarrollo de un estándar

La combinación del ciclo de vida del software y del proceso de estandarización de los organismos internacionales da lugar al proceso particular que siguió la presente investigación:

1. Análisis de requerimientos o definición del problema
2. Diseño o definición de una solución (bosquejo de la propuesta)
3. Revisión de la propuesta
4. Formulación del estándar

Análisis de requerimientos o definición del problema

Esta es una fase habitual del proceso de producción del software, durante la cual el ingeniero del software busca describir con precisión el problema que se le pide resolver y el dominio (negocio) al que pertenece. Es el resultado de la interacción con el cliente y otras fuentes (Davis, 1983).

Esta primera fase equivale a la que se lleva a cabo en los organismos estandarizadores, durante la cual surge la petición de satisfacer una necesidad definida mediante la creación de un estándar. En el caso presente, la necesidad es contar con un mecanismo estándar para

formular actividades de evaluación del aprendizaje para la educación virtual; el cliente que solicita una solución es un ente abstracto, a saber, el modelo pedagógico del elearning, cuyo objetivo es utilizar todo el potencial de internet para la educación, para ser capaz, en lo que a evaluación se refiere, de construir actividades de evaluación válidas, confiables, consistentes con sus principios pedagógicos.

El primer paso de la metodología fue, entonces, encontrarse con el “cliente” para delimitar los atributos más significativos, para esta investigación, de la evaluación del aprendizaje en la educación virtual. Esos atributos guiaron el descubrimiento y enumeración de necesidades informáticas. Finalmente el resultado fue un documento con el conjunto de requerimientos que se espera cumpla el producto final, el estándar.

Diseño y definición de una solución: Propuesta de especificación

En el ciclo de vida del software (Davis, 1983) esta etapa también se conoce como de diseño arquitectónico, porque se produce un modelo físico o una arquitectura definida del sistema. Aquí los requerimientos se traducen a componentes, interfases y funcionalidad. Este no es exactamente el caso de la presente investigación, aunque sí hay algún tipo de diseño, ya que debe concluirse en un modelo del estándar, en el que debe destacar, más que la funcionalidad, la estructura.

El resultado es un esquema de dicha estructura y la generación de la propuesta en un conjunto de XSD o esquemas de XML. Cada esquema puede presentarse de varias maneras, tanto textuales como gráficas. Para ir a tono con la práctica acostumbrada, se eligió utilizar UML, ya que es un lenguaje de modelado estándar capaz de expresar la estructura de un sistema, en combinación con el desarrollo en texto.

El modelo del estándar diseñado equivale a la propuesta que en los organismos estandarizadores se presenta para iniciar el proceso de revisión y comentarios.

Revisión de la propuesta

La tercera fase se ha tomado del proceso acostumbrado por los estandarizadores (Masie, 2002; véase Rehesaar, 1996). Con el fin de llevar a cabo un equivalente en pequeña escala, la propuesta resultante de la etapa de diseño, la fase 2, fue discutida en varios grados de profundidad por algunos pedagogos y expertos en ingeniería del software, entre ellos los asesores de la presente investigación, los desarrolladores de LMS de la Universidad de Montemorelos, el e42, y miembros de áreas de educación virtual de la misma universidad y otros centros de estudio. El objetivo fue enriquecer la propuesta con sus aportaciones y finalmente lograr el consenso a favor de ella.

La intervención de los expertos obligó a iterar el proceso, repitiendo los tres primeros pasos. En cada ciclo el esquema básico fue solamente refinado, conservando su formato, mientras los archivos XSD fueron cobrando forma.

Logrado el consenso, la propuesta estuvo en posición de arribar al último paso.

Formulación del estándar

En este punto las revisiones demostraron que la propuesta cumple con los requerimientos del modelo pedagógico y que la estructura normativa satisface las necesidades descubiertas. Por tanto, se publica la especificación a través de este documento de investigación.

Entregables

La fase culminante produce un estándar formado por los siguientes elementos conocidos como “entregables”: Documentación, código de los XSD y, opcionalmente, listas de revisión o *checklists*.

Documentación

Este es un conjunto de textos explicativos y diagramas que fundamentan o explican el estándar y sirven como guía para interpretar o aplicar los archivos XSD a documentos XML. El alcance y extensión que tienen es muy diversa, como puede verse en los documentos publicados por el IMS (véase IMS Global Learning Consortium, 2003), aunque procuran tener un formato básico. Además de la introducción o presentación de rigor, suelen incorporarse glosarios, referencias y contribuciones. Lo central lo constituye el desglose del estándar, textual o gráficamente.

Diagramas UML

La emergencia de las metodologías y lenguajes orientados a objetos han dado un importante empuje al lenguaje de modelado estándar del Object Management Group, el Unified Model Language (UML); este ofrece más de una decena de diagramas para expresar la estructura, funcionalidad, secuencias, jerarquías, interfases, etc. Si bien su utilidad gira alrededor del desarrollo de software, puede también recurrirse al UML para modelar otros proyectos. Uno de ellos es el modelado de los estándares (Object Management Group, 2004; Zavala, 2002).

Esquemas XML

El estándar propiamente consiste de varios esquemas de XML (XSD). Se consideró que estos eran más apropiados que los DTD, si bien ambos códigos sirven para validar las estructuras de datos de las aplicaciones que implementen el estándar. ¿En qué se diferencian? Los DTD surgieron antes, son más elementales y fáciles de crear, pero carecen de algunos elementos esenciales. Los esquemas incorporan muchos más tipos de datos primitivos y están expresados en XML; son más precisos para la función de validación que cumplen, pero aún no son soportados completamente (Griffin, 2002; Holzner, 2001).

Opcionalmente la documentación de los estándares puede incluir archivos XML de ejemplo que implementen los DTD o los esquemas.

Lista de revisión o checklist

Esta es una modalidad propia para la revisión de productos o procesos existentes, que consiste en la lista de ítemes que deben satisfacerse para afirmar que el estándar se ha cumplido. Cada ítem describe una cualidad esencial del estándar.

La prueba más directa del cumplimiento de la especificación que presenta esta investigación, es que los datos de evaluación del aprendizaje de un software educativo puedan ser expresados, sin transformaciones intermedias, mediante archivos XML bien formados y válidos, de acuerdo con los esquemas correspondientes. Y a la inversa, que dicho software sea capaz de interpretar directamente datos de evaluación que se le envían en XML validado.

En otras palabras, desde el punto de vista de un sistema, el código XML es suficiente para verificar el cumplimiento del estándar, así que una lista de revisión es aquí un elemento opcional. Sin embargo, también es útil producirla, por ejemplo, para administradores escolares o evaluadores de herramientas de elearning.

CAPÍTULO IV

PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

Este capítulo presenta la propuesta de estándar o especificación de software de evaluación del aprendizaje para la educación virtual. Siguiendo el proceso de cuatro fases expuesto en la metodología (ver capítulo III) aparece inicialmente el fundamento de esta especificación en la forma de un modelo pedagógico de evaluación. Este modelo se tradujo al conjunto de requerimientos que debe cumplir un software de evaluación del aprendizaje, para conformar lo que se conoce en ingeniería del software como modelo lógico, que no es más que la delimitación y esquematización del problema o necesidad a satisfacer.

Definido el problema, se pasó a la segunda fase, donde se diseñó la solución propuesta; el resultado es conocido en el ciclo de vida del software como modelo físico, ya que presenta de manera concreta la arquitectura del software solución, aunque aquí se trata no de modelar un software, sino una especificación de software. Cumplir el modelo físico es, en realidad, construir el estándar propiamente y elaborar los entregables: la documentación, el código y la lista de revisión que constituyen la propuesta de estándar.

El proceso de revisión por el cual pasó la propuesta de estándar, en cumplimiento de la tercera fase, está implícita en la presentación de estos resultados. La cuarta y última etapa del proceso es la publicación del estándar. Esta acción se materializa con la presentación de esta investigación.

Presuposiciones para describir e implementar los requerimientos

El modelo pedagógico es un bosquejo con sus elementos generales y se da por sentada su pertinencia. Los aspectos que sí exigen detalle son sus elementos relacionados con la evaluación del aprendizaje. La investigación presupuso que los elementos generales los aporta un LMS hipotético, sin considerar cómo son especificados. Por ejemplo, se utilizan objetivos de aprendizaje, sin detallar cómo se enuncian en un curso y de qué se componen, porque la propuesta de estándar presupone que a un software de evaluación del aprendizaje no le concierne crear objetivos de aprendizaje, le basta con hacer referencia a los que se han definido ya en el LMS, del que puede o no formar parte. Lo mismo puede decirse de los usuarios; la propuesta de estándar asume que basta con tener una referencia a ellos para relacionarlos con sus respuestas, resultados y calificaciones.

Por tanto, el lector notará que el detalle está puesto en los elementos que componen la evaluación del aprendizaje y que se deja independiente la implementación del LMS. No obstante, con el fin de dar claridad a la especificación se modelaron los elementos de un LMS que el subsistema evaluador requeriría.

Análisis de requerimientos o definición del problema

Objetivos. El propósito de un programa de estudio, denominado aquí genéricamente curso, es dar formación a los estudiantes, cumpliendo objetivos de aprendizaje definidos (Del Campo; Pérez, 1997; White, 1987). Por tanto, un curso consiste esencialmente en una secuencia diseñada de experiencias de aprendizajes, propuestas para cumplir los objetivos de aprendizaje de dicho curso en particular (Tattersall y Koper, 2003; véase Lafourcade, 1969).

Unidades de estudio. Las experiencias de aprendizaje ocurren durante y a través del desarrollo de unidades de estudio (también llamadas unidades de aprendizaje); los dos componentes de dichas unidades de estudio son los objetos de aprendizaje y el contexto; es decir, el contenido que se enseña y el marco en el que se enseña (IMS Learning Design Information Model, Version 1.0 Final Specification, 2003; Koper, 2001; Tattersall y Koper, 2003).

Objetos. Los objetos de aprendizaje son cualquier contenido autodefinido y reutilizable; sin un contexto y una relación con los objetivos de aprendizaje, estos objetos carecen de significado pedagógico (Tattersall y Koper, 2003). De acuerdo con la especificación para el empacamiento de contenidos del IMS, un objeto de aprendizaje puede estar formado a su vez por ítemes y subítemes, organizados en estructuras (IMS Global Learning Consortium, 2003).

Contexto. El contexto de un curso lo constituye el marco de trabajo en el que ocurre el proceso de enseñanza-aprendizaje; especifica los prerrequisitos (si los hay), el tiempo (duración, principio y fin), datos generales para evaluarlo (por ejemplo, la calificación mínima para aprobar) y otros datos más, que pueden agruparse en dos tipos de contexto: de uso y de evaluación; éste último puede considerarse opcional, ya que no siempre un curso culmina en una evaluación explícita.

Cada unidad de estudio tiene, por su parte, su propio contexto, derivado del contexto general del curso al que pertenece. Este contexto dicta dónde ubicar esta unidad, qué relación tiene con otras unidades y estructuras del curso y cómo puede accesarse a ella. Si además se le

dota de un contexto de evaluación, esta unidad se transforma en una actividad de evaluación que cumple explícitamente uno o más objetivos del curso.

Productos de aprendizaje. Tal como se vio antes, una actividad de evaluación es un evento del curso que tiene como fin verificar el logro de los objetivos de aprendizaje, durante o después de las experiencias de aprendizaje; esto es, durante o después del desarrollo de una o más unidades de estudio (véase Glabán y Ortega, 2002). Las actividades de evaluación pueden ser simples o compuestas de un conjunto de ítemes, como es el caso de las asignaciones que exigen la entrega de varios resultados, como los webquest o los exámenes (Dodge). El resultado que se busca con una actividad de evaluación en particular es uno de los siguientes: diagnóstico (actividad diagnóstica), formación o realimentación (actividad formativa) y calificación (actividad sumativa) (Cunningham, 1998; véase también Bloom, Madaus y Hastings, 1981).

De lo anterior se desprende que los productos de aprendizaje que se buscan son formación (realimentación) y, opcionalmente, calificaciones o un diagnóstico inicial.

Diagnóstico inicial: Sólo se obtiene con actividades aplicadas al principio de un curso, basadas en un patrón de prerrequisitos y cuyos resultados se relacionen con un estándar predefinido (Bloom, Madaus y Hastings, 1981).

Calificaciones: Obtenidas con actividades sumativas, que abarcan varias unidades de estudio y se aplican ocasionalmente a lo largo del curso (véase Bloom, Madaus y Hastings, 1981; College of Continuing Education, 2002). Cuando se busca una calificación, es necesario asociar la actividad con el esquema de evaluación del curso. El esquema de evaluación

expresa la valoración del logro de los objetivos de aprendizaje en términos de puntos, porcentajes o alguna otra escala equivalente.

La ponderación de una actividad y el cálculo de calificaciones en general pueden realizarse aplicando varias fórmulas de cómputo. La fórmula o combinación de ellas más apropiada es la que ofrezca resultados más reales, objetivos y útiles, y sea coherente con los objetivos de aprendizaje y las características de los estudiantes (Cunningham, 1998). Por otro lado, la interacción entre los actores del proceso educativo debe reflejarse en la evaluación, incorporando en algunas o todas las actividades elementos de colaboración, como la coevaluación y la autoevaluación (Wiggins, 1998).

Formación o realimentación: Su punto de partida son las actividades formativas, que evalúan porciones pequeñas del curso, son de corta duración y son aplicadas con mayor frecuencia que las anteriores (Wiggins, 1999; véase también Bloom, Madaus y Hastings, 1981). La realimentación es posible cuando el programa del curso y las actividades en general son desarrollados en forma flexible, para permitir adaptaciones según el progreso evidenciado. La adaptación de las actividades de evaluación puede darse en número, periodicidad, enunciación, configuración, peso en la calificación y secuencia.

Esquema de evaluación: Se ha dicho que cuando una unidad de estudio es evaluable, esto es, que se trata de una actividad de evaluación, es necesario asociarla con un esquema de evaluación definido. Este representa un convenio o contrato entre docente y alumno que describe a grandes rasgos los resultados cuantitativos que pueden conseguirse cumpliendo los requerimientos del curso. El esquema en sí es un conjunto de ítemes entre los que se divide la calificación final; cada ítem expresa en términos genéricos cómo debe ocurrir la evaluación;

son estos los que realmente se relacionan con una actividad de evaluación calificable. Un ítem de evaluación representa básicamente un tipo de actividades de evaluación o un segmento del curso. Cada curso tiene un esquema de evaluación y éste es desarrollado a partir de los objetivos de aprendizaje.

Con los conceptos vertidos hasta aquí es posible conformar el modelo pedagógico que se esquematiza en la Figura 4.

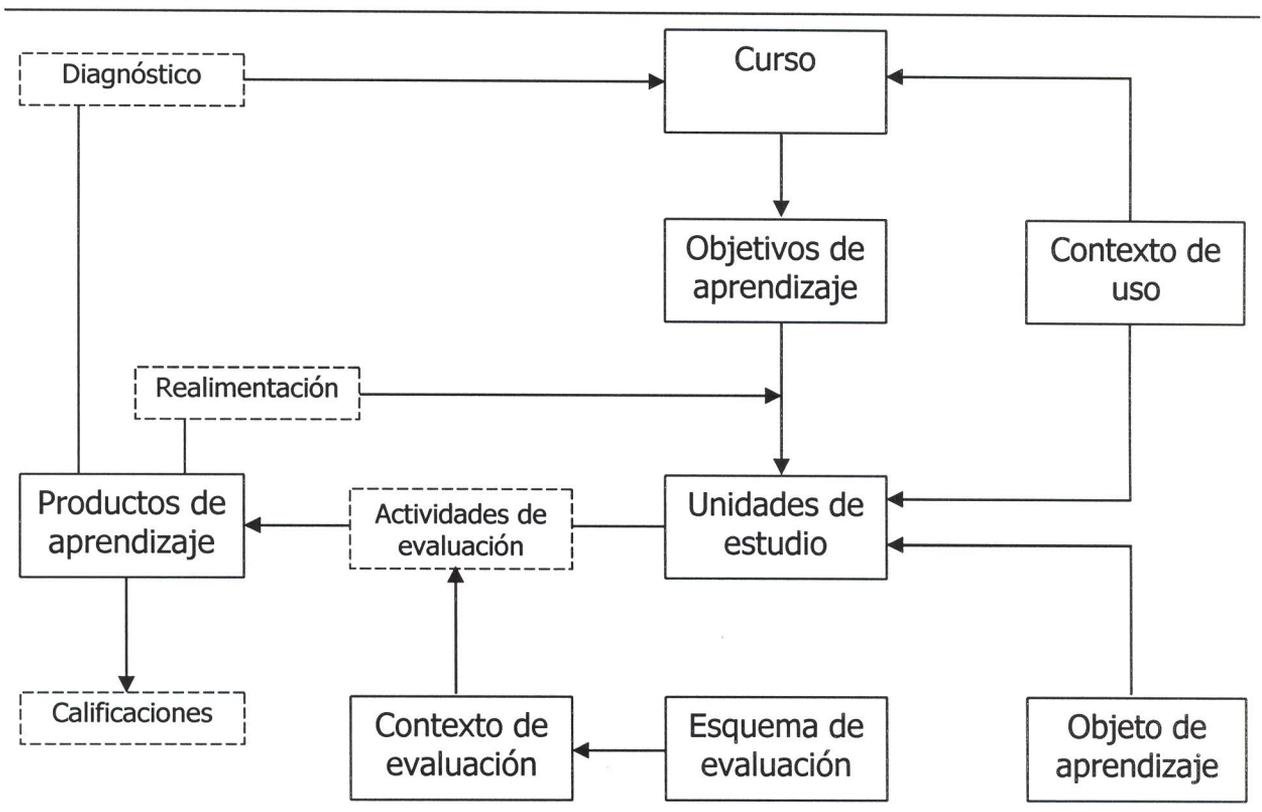


Figura 4. Esquema del modelo pedagógico de evaluación del aprendizaje

Obtención de requerimientos: Modelo lógico

Según lo visto, el sistema administrador del elearning debe seguir consistentemente un modelo pedagógico. Por tanto, un sistema o subsistema de evaluación del aprendizaje debe basarse en un modelo pedagógico de evaluación del aprendizaje. Se usa este argumento para

utilizar los requerimientos que *hace* el modelo a un sistema hipotético como los requerimientos que debe cumplir el estándar que busca esta investigación. Dichos requerimientos son los siguientes (se destacan elementos que en la siguiente fase se transforman en clases o componentes estructurales):

Requerimientos generales de un sistema de educación virtual

Estos requerimientos los cumple el LMS; se enuncian los que son significativos para el sistema evaluador.

Curso. Todo el proceso educativo debe ocurrir dentro de una entidad llamada Curso. El o los usuarios que dirigen dicho curso son los Instructores.

Objetivos de aprendizaje. El Curso define uno o más Objetivos de aprendizaje únicos. Los usuarios que buscan cumplir en ellos mismos dichos objetivos son los Alumnos; estos pueden estar organizados en estructuras que llamamos Grupos o equipos de trabajo. Hay siempre un grupo implícito que incluye a todos los alumnos de un curso.

Secciones. El Curso puede organizarse en una o más Secciones, y éstas a su vez subdividirse, opcionalmente, en otras secciones, en tantos niveles como se requiera. Las secciones son, entonces, estructuras que organizan el curso en un orden definido. Siempre hay al menos una sección implícita, que agrupa todo el curso.

Unidades de estudio. Los contenidos que se estudian en un curso se encuentran en las Unidades de estudio. Un curso tiene un número arbitrario de unidades, pero para que éstas se consideren como tales se componen siempre de un Objeto de aprendizaje, hacen referencia al cumplimiento de uno o más Objetivos, y se ubican en un Contexto. Cada unidad pertenece a

una sección del curso, y puede contener a su vez otras secciones con unidades de estudio secundarias. También puede estar compuesta de ítemes dependientes, organizados dentro de la unidad.

1. Los Objeto de aprendizaje son contenidos autodefinidos y reutilizables, por lo tanto pueden proceder o no del curso; esto quiere decir que son exportables e importables, pero sólo pueden relacionarse significativamente con un Curso cuando se encuentran en un contexto, formando parte de una Unidad de estudio.

2. El Contexto es el conjunto de datos que enmarcan todo un curso o partes de él. Incluye las especificaciones de Tiempo y las Relaciones entre elementos del curso, formando un Contexto de uso. Los elementos evaluables (el curso en sí, las unidades de estudio y sus ítemes) poseen además un Contexto de evaluación, constituido por las especificaciones de la Aplicación de la evaluación y por la mecánica para realizar el Cómputo de resultados.

3. Las especificaciones de Tiempo establecen el inicio y fin del curso, en términos de fechas concretas o de un período medido en días.

4. Las especificaciones de Relación establecen la ubicación e interrelaciones de todos los elementos del curso y además permiten definir, opcionalmente, Secuencias de elementos, de modo que el acceso o el peso de un elemento del curso dependan de un comportamiento o logro demostrado en una unidad o conjunto de unidades de estudio previo.

Registro del alumno. Cada alumno del curso debe poseer un registro de su relación con dicho curso, que indique cuándo ingresó, cuándo egresó, cuál es su status actual y qué calificación ha recibido, si se trata de un curso evaluable.

Requerimientos de un subsistema de evaluación del aprendizaje en la educación virtual

Este subsistema debe ofrecer al curso básicamente tres componentes: Un Contexto de evaluación, para convertir los elementos del curso en evaluables, un controlador de Respuestas y Calificaciones, y los Reportes correspondientes a estas últimas.

Contexto de evaluación. En el Contexto de evaluación deben definirse las especificaciones para la Aplicación de las actividades: si lo obtenido por un alumno al resolver una actividad cuenta para su calificación global y, en caso afirmativo, cuánto debe contar; si el alumno puede autoevaluarse y si sus compañeros pueden evaluarlo; cómo se accederá a la revisión de respuestas; y qué tipo genérico de actividad se está implementando. Además, en este contexto debe definirse el modo de realizar el Cómputo de los resultados: cuánto valor (en puntos o porcentaje) se obtiene al resolver una actividad; si dicho valor debe distribuirse automática o manualmente entre los ítemes que componen una actividad, si los hubiera; y qué fórmula se aplicará al ponderar los resultados de una actividad o conjunto de ellas.

1. Cada tipo genérico de actividad de evaluación lo define un Ítem de evaluación.

Estos tipos o ítemes establecen si una evaluación es objetiva (cuando se evalúa una actividad realizada por un alumno), subjetiva (cuando el instructor evalúa un comportamiento o actitud sin que medie una actividad enviada) o de votación (cuando el logro de una actividad depende de los votos que ha recibido, en una mecánica parecida a las encuestas rápidas). Además, los ítemes deben indicar qué porcentaje o cuántos puntos de la calificación global se obtienen al cumplir las actividades que lo implementan.

2. Las Fórmulas de cómputo son arbitrarias, pudiendo el software para el elearning incluir algoritmos que implementen, por ejemplo, la búsqueda de promedios, de desviaciones

estándar, de puntuaciones z, etc. Cuando no se define una fórmula de cómputo para una actividad o conjunto de actividades, se asume que debe tomarse la calificación bruta obtenida.

Calificaciones. Las Calificaciones se componen de la colección de Respuestas de un alumno a una actividad, la suma de resultados objetivos, subjetivos y de votación y los puntos extras que el instructor añade a la calificación global del alumno. Esta calificación global más los puntos extras constituirá la calificación final del alumno. El rango de puntos extras posibles depende de la configuración del LMS.

1. Las Respuestas se forman por este conjunto de datos: cuándo fue enviada, el texto o el objeto que conforman la respuesta, cuándo fue revisada por el instructor y el texto u objeto que conforman la revisión. Además, incluyen la calificación que el instructor asigna, independientemente de que cuente o no para la calificación final; la calificación que el propio alumno se asigna, si está habilitada la autoevaluación; y el promedio de calificaciones que asignan los compañeros, si está habilitada la coevaluación. Finalmente, el sistema genera un status para cada respuesta (enviada, revisada, rechazada o reprogramada para repetirse).

2. Otro conjunto de evaluaciones son las que realizan los alumnos a la respuesta de un compañero. Si se trata de una actividad de evaluación objetiva, los alumnos deben poder incluir una calificación y comentar la respuesta de su compañero; si es una actividad de votación, la evaluación consiste en votar eligiendo de entre una serie de valores que el sistema ofrezca (por ejemplo, calificar del 1 al 10). El número de opciones y el tipo de valor es arbitrario, con tal de que produzcan resultados cuantitativos.

Reportes. Los Reportes constituyen la presentación visual del proceso de evaluación que realiza el sistema evaluador, para que los involucrados cuenten con la información para

realimentar el curso. El proceso de realimentación debe incluir la capacidad de modificar los ítemes de evaluación y el contexto de evaluación de una actividad, y de especificar o modificar una secuencia de eventos respecto a cierta actividad. Incluso las respuestas pueden no ser inmutables, si se habilita la opción de que cierta actividad sea repetible.

Diseño arquitectónico o definición de la solución: El Estándar

A continuación se presenta un estándar para software de evaluación del aprendizaje que satisfaga los requerimientos enunciados anteriormente.

La primera aproximación la constituye un diagrama de casos de uso que ilustra a grandes rasgos los elementos involucrados (Figura. 5). Tal como lo especifica el UML, los actores (Usuario y Objeto de aprendizaje) constituyen elementos ajenos al sistema, aun cuando se da el caso de que el propio LMS crea esos objetos; la intención de verlos como actores es destacar que son elementos independientes del proceso de enseñanza-aprendizaje (un usuario podría existir para una escuela, por ejemplo, aunque en ese momento no tome clases). También se observa que Instructor y Alumno son una especialización de la clase Usuario; igualmente la Actividad de evaluación es una especialización de Unidad de estudio. Se deja al implementador la manera de aplicar este hecho, siendo una opción utilizar herencia. Como se verá más adelante, en el esquema de XML, la especialización se maneja con la palabra clave *extension* (extensión).

La Figura 6 presenta el refinamiento de los casos de uso en la forma de un diagrama de clases. Dada la naturaleza del estándar, no se intenta ofrecer una propuesta funcional, sino estructural. El diagrama de clases incluye todas las clases involucradas en el sistema evaluador. Las que proceden del LMS se han limitado a las necesarias para el evaluador.

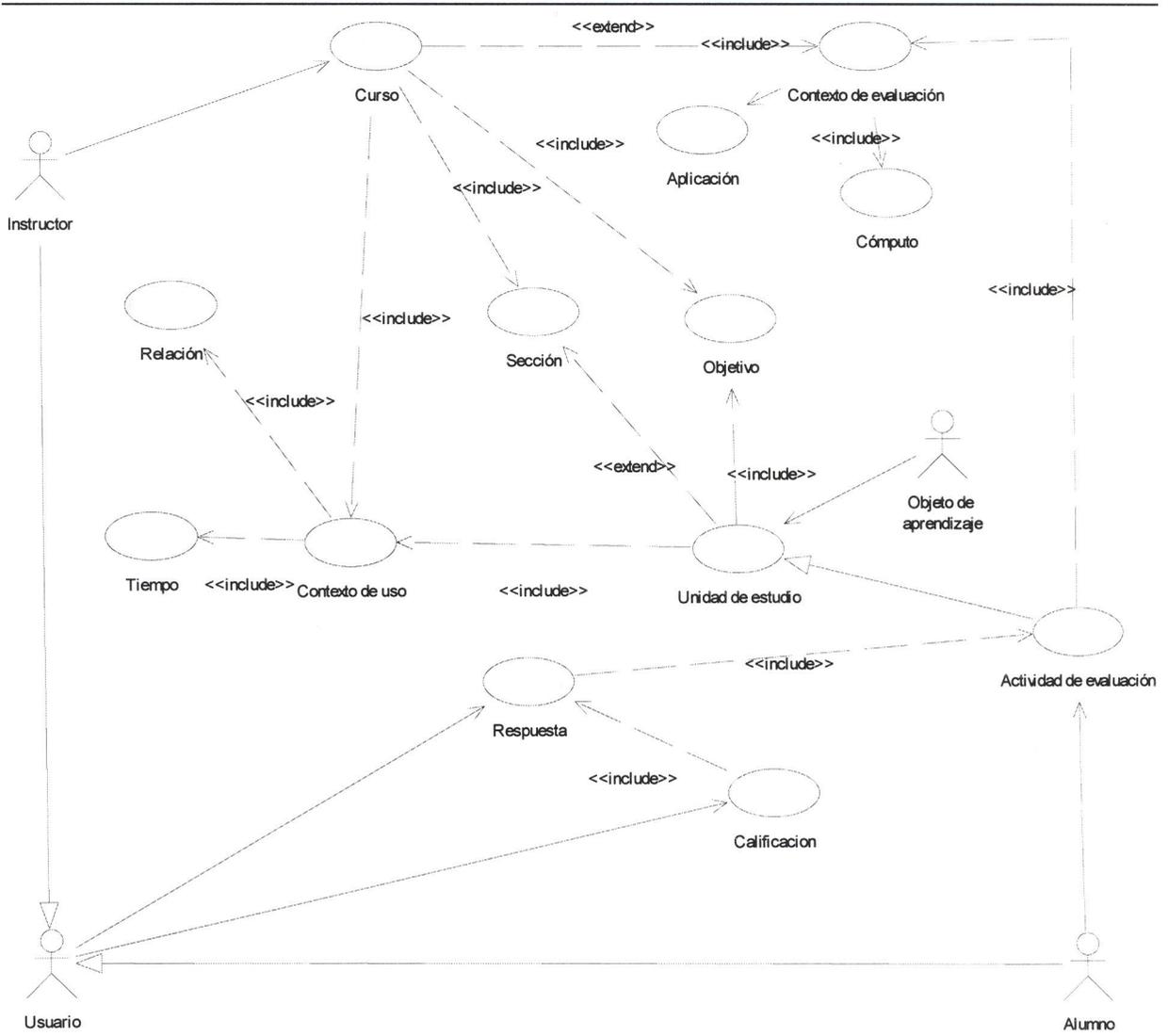


Figura 5. Casos de uso del estándar

Se proponen algunas interfaces para los comportamientos comunes a varias clases. Por ejemplo, la interfase I Estructura es la base para definir la ubicación y la relación de los elementos del curso. El contexto de uso y el contexto de evaluación están implícitos; el primero se forma implementando la interfase de tiempo, ITiempo, y la de relación, IRelacion, la cual incluye la clase Secuencia. El contexto de evaluación es la suma de las interfaces de

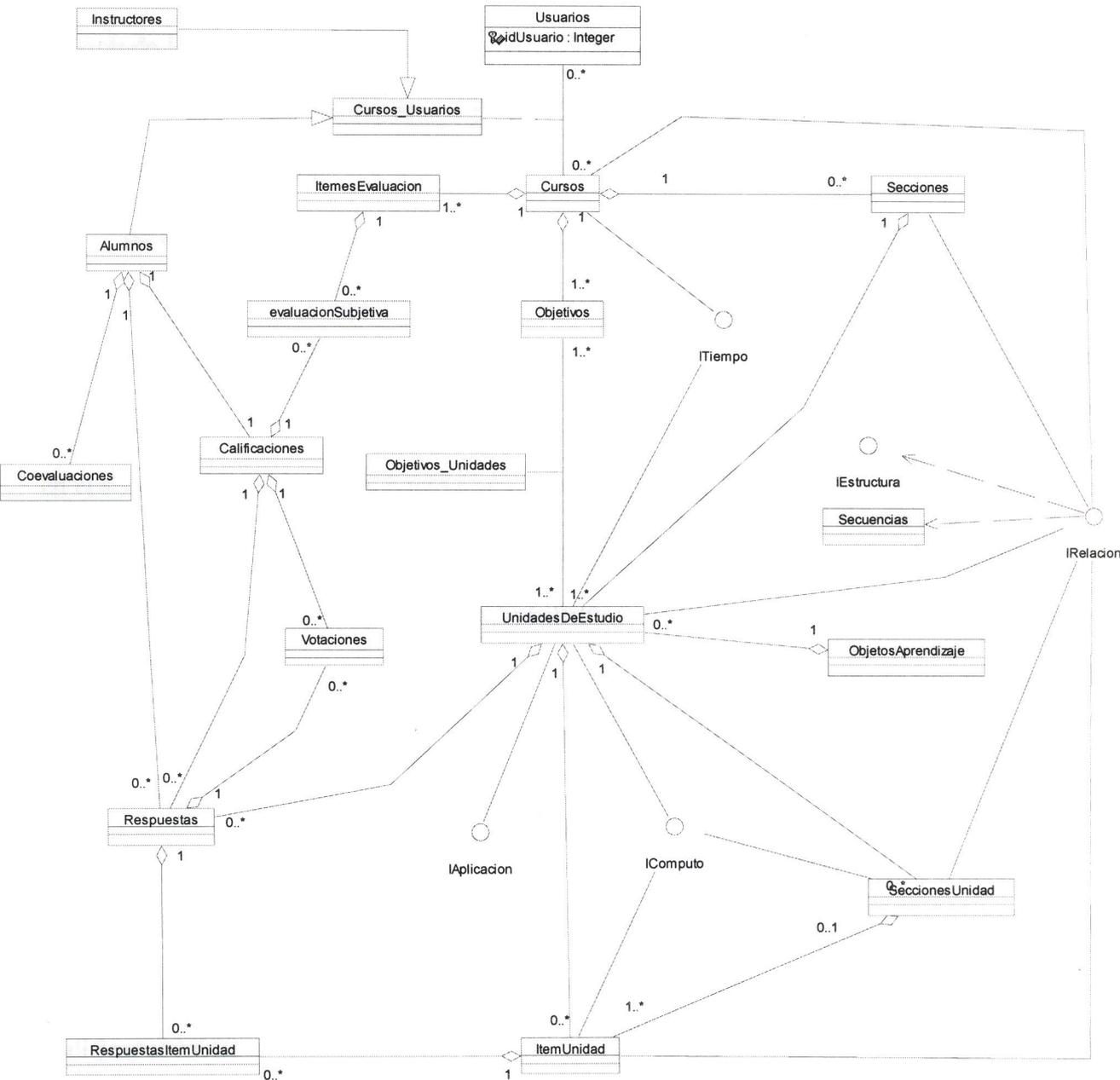


Figura 6. Diagrama de clases de UML de los objetos que componen un LMS

cómputo, IComputo, que incluye la clase Formulas, y la de aplicación, IAplicacion, que incluye la clase ItemsEvaluacion.

La Figura 7 es el mismo diagrama de clases representado en la Figura 6, pero mostrando los atributos de las clases. Se observa que ninguna clase incluye métodos dado que, como se mencionó, el estándar es una propuesta estructural y no funcional. La funcionalidad se deja al implementador.

Se designado un identificador para cada objeto que tenga que reverenciarse inequívocamente. Podría definirse como de tipo ID, que es en realidad un tipo del XML, normalmente interpretado como un String, pero se deja a la implementación la posibilidad de utilizar otros tipos primitivos, con tal de conservar la cualidad de único de ese valor del objeto en el sistema. Otra posibilidad hubiera sido utilizar el tipo Key del XML, que puede ser más flexible, aunque el ID se aplica a identificadores globales. La presente propuesta de estándar utiliza el tipo entero (integer) como implementador del ID y del IDRef, ambos identificadores.

En una base de datos, este ID equivaldría a la llave primaria o *primary key*, mientras que para las llaves externas o *foreign key* se utilizaría el tipo IDRef. Se implica así que tanto ID como IDRef pueden formarse con más de un valor.

Las relaciones y la cardinalidad de las relaciones entre las clases también se expresan en los XSD, pero se aprovecha que un diagrama de clases tiene la virtud de ser más evidente y más conocido que aquéllos, por lo que la secuencia a seguir pasó primero por el UML antes de codificar el XSD.

Los requerimientos enunciados al principio del capítulo y el diseño arquitectónico representado por los diagramas de UML fueron finalmente traducidos a los esquemas de XML, los cuales constituyen en sí el estándar. Para facilitar la codificación y su uso el estándar se dividió en ocho archivos XSD. Los correspondientes a Usuarios y Objetos de aprendizaje enuncian escuetamente estas clases, destacando especialmente que poseen un

identificador. Se han separado para mostrar que representan elementos ajenos no sólo al evaluador sino al curso; de esta manera pueden practicarse diferentes implementaciones de Usuarios y Objetos de aprendizaje, sin afectar el LMS ni el evaluador.

También se encuentra separado el conjunto de interfases. El XSD correspondiente sólo contiene una, IEstructura, equivalente al IEstructura del diagrama de clases UML. La intención fue prever la implementación de otras interfases para el curso y centralizar su ubicación.

El cuarto XSD implementa el contexto de uso. Este es en realidad un ensayo que no pertenece realmente a la propuesta de estándar, ya que puede sustituirse por otras implementaciones. De hecho, el elemento Secuencia puede verse como una extensión del secuenciador de las especificaciones del IMS. Sin embargo, si un LMS no tuviera estos elementos contextuales, puede utilizarse este esquema que está completo para ello.

El XSD para las Unidades de estudio se considera parcialmente parte de la propuesta de estándar, en la medida en que las unidades pueden *transformarse* en actividades de evaluación al incluirse un contexto de evaluación. De modo que sólo son completamente parte del sistema evaluador los XSD correspondientes al Contexto de evaluación y a las Calificaciones, que incluye la especificación para las respuestas. Esto quiere decir que, excepto estos dos esquemas, todos los demás pueden ser sustituidos por implementaciones particulares sin que se afecte la lógica del modelo en lo tocante a la de evaluación.

El último XSD, Cursos, presenta un modo de reunir todos los elementos. También pertenece estrictamente al LMS, pero conviene haberlo modelado para comprender mejor los requerimientos que le haría a un subsistema evaluador y, sobre todo para tener una idea más clara de la interrelación entre el evaluador y el resto del sistema.

Entregables

En el Apéndice A se encontrará el código completo de los ocho XSD.

La documentación de los esquemas, que forma parte de los entregables, se hallará en el Apéndice B. Su finalidad será describir todos los elementos y atributos de los esquemas, su relación con otros esquemas y atributos, así como consideraciones acerca de la implementación. Además, el Apéndice C presenta el ejemplo de los datos de evaluación de un curso, que han sido vertidos a un archivo XML validado por los XSD del estándar, y el Apéndice D ofrece una lista de recursos web recomendables para utilizar los esquemas de XML, así como la descripción del contenido del CD anexo.

A pesar de la explicación y los ejemplos, no es difícil descubrir que el mero código no transmite expresamente los requerimientos pedagógicos que debiera cumplir el sistema. Indudablemente sigue siendo una necesidad poseer una descripción, en lenguaje llano, de cualquier sistema. Precisamente para el caso de los estándares una modalidad muy utilizada son las listas de revisión que, aunque se trataran de un documento técnico, ofrecen la ventaja de articularse en lenguaje ordinario.

Sin embargo, las listas de revisión no definen con la precisión del código la estructura de datos y las relaciones entre objetos; con todo, puede resultar útil para evaluar en forma general software existente, o como una guía de desarrollo. Así que a continuación se enuncia la propuesta de estándar para evaluación del aprendizaje en la educación virtual, en la forma de una lista de revisión. Esta lista puede adoptar varios estilos, ser una simple *checklist*, una serie de cualidades que inicien con la palabra “debe”, una lista de ítemes calificables, etc. Aquí se enuncia sencillamente como una lista numerada de ítemes.

Nuevamente, para juzgar la funcionalidad y propiedades del sistema evaluador, se da por sentado que el LMS ofrece los siguientes elementos (los cuales, por tanto, no se incluyen entre los ítemes a revisar):

Constructor de cursos; registro de usuarios con asignación de roles (al menos identificación de instructores y alumnos); editor de objetivos de aprendizajes del curso, relacionados opcionalmente con un conjunto de competencias (habilidades y actitudes) que se pretenden desarrollar al alcanzar los objetivos de aprendizaje; organizador del curso en estructuras (secciones); definidor de un contexto de uso general que establece la duración del curso, su inicio y final, quiénes pueden accederlo, que prerequisites exige, qué servicios o singularidades posee y con qué otros recursos del sistema se relaciona; creador o importador de objetos de aprendizaje o contenidos, para formar unidades de estudio.

Teniendo los elementos anteriores, aportados por el LMS, se enlistan los elementos que caracterizan a un subsistema de evaluación del aprendizaje, según la propuesta de estándar plasmada en los esquemas de XML.

Lista de revisión

El sistema evaluador:

1. Define un esquema de evaluación único para el curso, compuesto por uno o más ítemes de evaluación
2. Los ítemes de evaluación definen la cantidad de puntos o el porcentaje de la calificación general que se distribuirán entre las actividades de evaluación o las secciones del curso
3. Define secuencias reutilizables para exponer, ocultar, diferir u omitir contenidos y actividades, basadas en el desempeño de un alumno o grupos de alumnos

4. Provee fórmulas de cómputo que pueden utilizarse para calcular calificaciones a partir de resultados de los alumnos

5. Posee un repositorio de las respuestas de los alumnos a las actividades de evaluación

6. Maneja calificaciones parciales y finales de los alumnos, que son el resultado de aplicar las fórmulas de cómputo elegidas a las respuestas de los alumnos a las actividades de evaluación

7. Ofrece un resultado final conformado por la calificación final, el status formal que el alumno ha alcanzado al final del curso y las competencias que ha alcanzado de acuerdo con los objetivos cumplidos

Cada actividad de evaluación tiene estas propiedades:

8. Ocurre en el contexto de un curso, programa de estudio o programa de evaluación diagnóstica; por lo tanto pueden definírsele la fecha y la forma de acceso, así como la secuencia que sigue respecto a otras actividades de evaluación

9. Se relaciona con uno o más objetivos de aprendizaje del curso

10. La conforma un objeto de aprendizaje o contenido y, si existen, los ítemes y subítemes de dicho objeto

11. Posee un valor, en puntos o porcentaje, y puede definir si cuenta o no para la calificación del alumno, y en caso positivo en qué proporción contaría, para lo cual se relaciona con un ítem del esquema de evaluación

12. Puede definir qué fórmula de cómputo debe aplicarse cuando es resuelta

13. Puede habilitar o deshabilitar la autoevaluación

14. Puede habilitar o deshabilitar la coevaluación

15. Posee mecanismos para distribuir manual o automáticamente su valor o puntos entre sus secciones e ítemes, si existen

16. Puede poseer secciones para organizar los ítemes que la componen, si existen

CAPÍTULO V

CONCLUSIONES

La gran pregunta que queda por responder en esta investigación es qué utilidad tiene realmente la propuesta de estándar que se ha elaborado.

Ya ha sido expuesto que la educación virtual llegó para quedarse; que ya no se discute si es viable o si superará la novedad, sino cómo se ha de implementar, cómo justificarla pedagógicamente y cómo aprovecharla. También se ha argumentado que las características de internet (contenidos no lineales, distribuidos, facilidad para la interrelación, inmediatez y ubicuidad) están favoreciendo precisamente las modernas corrientes pedagógicas que, entre otras cosas, comparten esas mismas características. Y aunque los detractores de la estandarización en la ingeniería del software tienen razón en que hay muchos hilos sueltos en el dominio de esta joven disciplina, a estas alturas debe haber quedado claro que los estándares de la ingeniería del software representan un componente obligado en la unión de la educación y la informática, especialmente en la educación virtual.

Sin embargo, en ese marco ¿es significativa la aportación que pretende esta propuesta de estándar? Esto no puede con certeza responderse sino hasta que el estándar se utilice ampliamente y resuelva problemas reales. No obstante, es posible obtener una vislumbre del su alcance al revisar cómo cumple los objetivos de esta investigación (véase el capítulo introductorio).

Cumplimiento de objetivos

1. La propuesta de estándar cumple los requerimientos del modelo pedagógico encontrado, ya que se basó precisamente en tales requerimientos; además siguió, para su creación, un proceso surgido de los procesos de las grandes instituciones estandarizadoras.

2. Los esquemas de XML elaborados pueden aplicarse para certificar la calidad del software educativo para evaluación del aprendizaje; si los datos producidos por dicho software son validados por los esquemas, se afirma que el estándar se ha cumplido.

3. La lista de revisión derivada puede utilizarse para evaluar de manera general el software educativo para evaluación del aprendizaje. Si este software muestra que es capaz de producir u ofrecer los requerimientos de la lista de revisión, el estándar se ha cumplido.

4. Las listas de requerimientos y los diagramas pueden servir de guía al crear nuevo software. Diversas herramientas en el mercado pueden incluso crear una base de datos a partir de esquemas de XML. Además, la manera en que se han presentado los esquemas de la propuesta de estándar favorece el desarrollo por componentes, independientes y reutilizables. En otras palabras, aquí está ya adelantada la fase de requerimientos, buena parte del diseño y lo central de la base de datos.

5. Como se espera de los estándares de la ingeniería del software en general, la documentación presentada puede facilitar la comunicación entre quienes se relacionan con este tipo de software: ingenieros, productores, pedagogos y clientes, ya que ofrece una terminología común, describe las propiedades del software textual y gráficamente, y define claramente el producto esperado en términos suficientemente amplios. Este documento hace posible, por ejemplo, que un docente y un ingeniero puedan entenderse más fácilmente al discutir sobre evaluación del aprendizaje y el software relacionado.

No sólo en el campo del desarrollo y la evaluación de la calidad hay aplicación para esta especificación. El autor de esta investigación sugiere que otra aplicación importante puede hallarse en la interoperabilidad de sistemas heterogéneos. Una preocupación constante de las empresas ha sido la de aprovechar sus sistemas heredados (*legacy*) e integrarlos con sistemas de reciente adquisición, o la de acoplar sistemas desarrollados con diferentes tecnologías. Uno de los elementos que forma parte de las soluciones más modernas es el uso de XML, aplicado hoy en día tanto en sistemas distribuidos como en sistemas de escritorio. Así que este estándar, formulado en XML, ofrece una estructura que puede ser aprovechada en instituciones educativas o capacitadoras que poseen sistemas que no se comunican directamente, pero a los que puede añadirse una capa de comunicación de la evaluación del aprendizaje basada en el estándar (Figura 8).

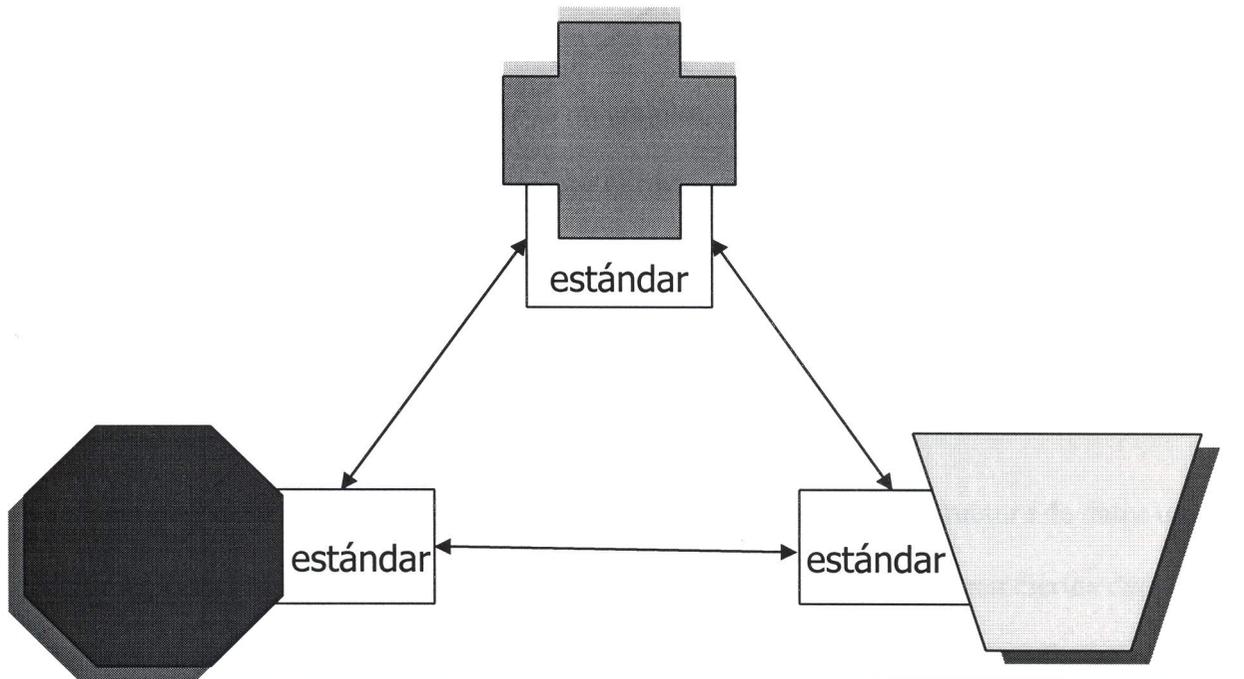


Figura 8. Interoperabilidad entre sistemas heterogéneos mediante el estándar

Implementación del estándar

Queda por resolver, finalmente, cómo se implementa el estándar. Dada su naturaleza, en realidad se trata de establecer una estructura de datos, independientemente de su aplicación. En otras palabras, si se quiere desarrollar un software de evaluación del aprendizaje basado en los esquemas XML expuestos, se indican los siguientes pasos:

Paso 1. El LMS

Como se dijo en los capítulos anteriores, se considera que el software de evaluación del aprendizaje es subsistema de un LMS existente. Por tanto, el desarrollador debe tener los datos básicos o referencias a los elementos del LMS, los cuales se ilustran en los esquemas XML auxiliares, según la documentación del capítulo IV. Ahí, por ejemplo, se indica que el evaluador produce calificaciones para un curso; este curso se referencia con un identificador de tipo *integer*. El estándar no indica cómo debe obtenerse este identificador, con tal de que sea válido (de un curso existente, único y de tipo entero). Puede ser leído de una base de datos, recibido como variable de sesión, leído de un archivo, etc.

Todos los datos previos ofrecidos por el LMS están descritos en la documentación de los esquemas Cursos, Interfases, Usuarios, ObjetosAprendizaje, UnidadesEstudio y ContextoUso.

Paso 2. Estructura de datos del evaluador

La propuesta de estándar es en realidad la descripción de una estructura de datos que define que, para registrar los eventos de evaluación en un curso, se requieren ciertos datos, con un tipo definido, con cierta cardinalidad y relacionados de un modo específico. Se subraya el hecho de que los esquemas no establecen cómo debe implementarse esa estructura. Sin

embargo, existen básicamente cuatro formas de desarrollarla a partir de los esquemas propios del estándar, ContextosEvaluacion y Calificaciones:

El estándar en el código. Siguiendo la documentación de los esquemas, cada elemento o dato se convierte en una variable en el código. Usando programación orientada a objetos, los elementos mayores pueden transformarse en objetos, usando los datos de los esquemas como sus atributos. Por ejemplo, pueden implementarse las clases Calificaciones, Respuestas, ItemEvaluacion, etc.; este último tendría como atributos idItemEvaluacion, nombre, valor, tipoValor y tipoEvaluacion. Para cada clase se crearían los métodos necesarios; algunos pueden ser obvios, como constructores, validadores de tipos de datos y ordenadores de ítems; otros métodos pueden ser específicos del sistema que se intenta construir.

Con esta forma, el almacenamiento persistente de los datos no es crucial. De hecho, el repositorio de datos podría no coincidir en absoluto con la estructura del código, con tal de que se creen métodos para transformar los datos físicos en una estructura virtual, que satisfaga el estándar, creada cuando sea requerido. ¿Qué caso podría elegir esta vía? Posiblemente un sistema existente con un repositorio de datos ya establecido, posiblemente grande, que no valga la pena migrar.

El estándar en el repositorio de datos. En esta segunda forma, la estructura de datos física sigue al estándar. Sea una base de datos, archivos de texto, archivos de XML u cualquier otra forma de persistencia de datos, el repositorio se construye con base en los esquemas del evaluador, según la documentación. En cambio el código puede tener una estructura lógica diferente, aunque también requeriría un conjunto de métodos para transformar los datos lógicos en los datos físicos estandarizados.

Este podría ser el caso de un sistema existente que tiene otras funcionalidades, y para el que es más fácil crear un repositorio de datos alternos y un conjunto de métodos transformadores, antes que cambiar todo el código.

El estándar en el mecanismo transformador. Puede ocurrir que no convenga modificar ni el código ni el repositorio de datos. La alternativa aquí es agregar mecanismos para transformar los datos actuales en otros que satisfagan el estándar, bien sea creando un conjunto de métodos en el código para producir datos virtuales (por ejemplo, XML *on demand*) o, si se cuenta con un *data base management system* (DBMS), pueden programarse procedimientos en la base de datos para crear archivos XML o bien, ofrecer vistas con los datos transformados.

El estándar en el código y en el repositorio de datos. La última alternativa es una combinación de las dos primeras, recomendada para los sistemas nuevos o para aquellos para los que es menos costoso adaptarse completamente que conservar algo de su estructura y funcionalidad original. En esta forma el código incluye elementos (clases, por ejemplo) que coinciden con los elementos mayores de los esquemas XML, teniendo como atributos los datos de dichos esquemas. Y además, el repositorio de datos sigue la misma estructura, de modo que no se requerirían más que métodos de lectura y escritura directos.

En cualquiera de los casos, dado que se busca la interoperabilidad, se recomienda implementar métodos para interconectarse con otros sistemas. Lo natural, por supuesto, será la importación y exportación de datos en archivos XML, lo cual implica crear métodos que los validen contra los esquemas dados. Opcionalmente pueden crearse método para manejar XSLT.

Perspectivas de desarrollo

Y para concluir, una mirada a lo que sigue: Se sugiere estar atentos a la evolución de los estándares del IMS y al desarrollo del SCORM. Su revisión seguramente inspirará mejoras para siguientes versiones de este estándar. Adicionalmente se sugieren proyectos e investigaciones que se derivan de esta propuesta:

1. Desarrollo de software de evaluación diagnóstica, por ejemplo para aplicar exámenes de admisión o para evaluar la evolución del desempeño.
2. Desarrollo de los requerimientos de un software para investigaciones estadísticas en línea; por ejemplo, para aplicar encuestas complejas por internet y obtener resultados por el mismo medio.
3. Desarrollo de un empaquetador de objetos de aprendizaje, según los estándares ya elaborados por el IMS y otros, para intercambio de recursos entre instituciones educativas.
4. Investigación sobre el impacto de los LMS sobre los hábitos de navegación, estudio, comunicación y la autorregulación.
5. Comparación de los resultados de las actividades de evaluación del aprendizaje entre el ambiente presencial y el virtual, al utilizar sistemas basados en la propuesta aquí presentada.

APÉNDICE A

CÓDIGO FUENTE DE LOS ESQUEMAS XML

Cursos.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="Usuarios.xsd"/>
  <xs:include schemaLocation="ContextosUso.xsd"/>
  <xs:include schemaLocation="UnidadesEstudio.xsd"/>
  <xs:include schemaLocation="ContextosEvaluacion.xsd"/>
  <xs:include schemaLocation="Calificaciones.xsd"/>
  <xs:element name="Curso">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="idCurso" type="xs:integer"/>
        <xs:element name="datosCurso">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="relacion" type="InterfaseRelacion"/>
              <xs:element name="tiempo" type="InterfaseTiempo"/>
              <xs:element name="puntajeMaximo" type="xs:integer"/>
              <xs:element name="puntajeMinimo" type="xs:integer"/>
              <xs:element name="puntajeMinimoParaAprobar" type="xs:integer"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="instructor" type="UsuarioType" maxOccurs="unbounded"/>
        <xs:element name="alumno" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="UsuarioType">
                <xs:sequence>
                  <xs:element name="fechaIngreso" type="xs:date"/>
                  <xs:element name="fechaEgreso" type="xs:date" minOccurs="0"/>
                  <xs:element name="status">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="Solicitando"/>
                        <xs:enumeration value="Inscrito"/>
                        <xs:enumeration value="Rechazado"/>
                        <xs:enumeration value="Aprobado"/>
                        <xs:enumeration value="Reprobado"/>
                        <xs:enumeration value="Condicional"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element ref="Calificacion"/>
                </xs:sequence>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element ref="Objetivo" maxOccurs="unbounded"/>
        <xs:element ref="Seccion" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:element name="EsquemaEvaluacion" type="EsquemaEvaluacion"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Objetivo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="idObjetivo" type="xs:integer"/>
            <xs:element name="descripcion" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Seccion">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="idSeccion" type="xs:integer"/>
            <xs:element name="relacion" type="InterfaseRelacion"/>
            <xs:element name="tiempo" type="InterfaseTiempo"/>
            <xs:element ref="UnidadDeEstudio" maxOccurs="unbounded"/>
            <xs:element ref="Seccion" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

Interfases.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="IEstructura">
        <xs:sequence>
            <xs:element name="idEstructuraSuperior" type="xs:integer" minOccurs="0">
                </xs:element>
            <xs:element name="orden">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="nombre" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

Usuarios.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="UsuarioType">
        <xs:sequence>
            <xs:element name="idUsuario" type="xs:integer"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

ObjetosAprendizaje.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ObjetoType">
    <xs:sequence>
      <xs:element name="idObjetoAprendizaje" type="xs:integer"/>
      <xs:element name="itemObjeto" type="ItemObjetoType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ItemObjetoType">
    <xs:sequence>
      <xs:element name="idItemObjetoAprendizaje" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

UnidadesEstudio.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="ContextosUso.xsd"/>
  <xs:include schemaLocation="ContextosEvaluacion.xsd"/>
  <xs:element name="UnidadDeEstudio">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="idUnidadDeEstudio" type="xs:integer"/>
        <xs:element name="idObjetivo" type="xs:integer" maxOccurs="unbounded"/>
        <xs:element name="idObjetoAprendizaje" type="xs:integer"/>
        <xs:element name="relacion" type="InterfaseRelacion"/>
        <xs:element name="tiempo" type="InterfaseTiempo"/>
        <xs:element name="aplicacionEvaluacion" type="InterfaseAplicacion" minOccurs="0"/>
      </xs:sequence>
      <xs:element name="computo" type="InterfaseComputo" minOccurs="0"/>
    </xs:complexType>
    <xs:element name="seccionUnidad" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="idSeccionUnidad" type="xs:integer"/>
          <xs:element name="relacion" type="InterfaseRelacion"/>
          <xs:element name="computo" type="InterfaseComputo"/>
          <xs:element name="itemUnidad" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="idItemUnidad" type="xs:integer"/>
                <xs:element name="idItemObjetoAprendizaje" type="xs:integer"/>
                <xs:element name="relacion" type="InterfaseRelacion"/>
                <xs:element name="computo" type="InterfaseComputo"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:element>
<xs:element ref="UnidadDeEstudio" minOccurs="0" maxOccurs="unbounded"/>
</xs:schema>
```

ContextosUso.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="Interfases.xsd"/>
  <xs:complexType name="InterfaseTiempo">
    <xs:sequence>
      <xs:element name="esFechaAbierta" type="xs:boolean"/>
    </xs:element>
      <xs:element name="fechaInicio" type="xs:date" minOccurs="0"/>
      <xs:element name="fechaFin" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:date"/>
        </xs:simpleType>
      </xs:element>
      <xs:element name="duracion" type="xs:float" minOccurs="0"/>
      <xs:element name="unidadDuracion" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Mes"/>
            <xs:enumeration value="Dia"/>
            <xs:enumeration value="Hora"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="InterfaseRelacion">
    <xs:sequence>
      <xs:element name="ubicacion" type="IEstructura"/>
      <xs:element ref="Secuencia" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Secuencia">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="idSecuencia" type="xs:integer"/>
        <xs:element name="recursoPrevio" type="xs:integer"/>
        <xs:element name="tipoRecursoPrevio">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Curso"/>
              <xs:enumeration value="Seccion"/>
              <xs:enumeration value="Unidad"/>
              <xs:enumeration value="SeccionUnidad"/>
              <xs:enumeration value="ItemUnidad"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="cardinalidadRecursoPrevio">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Item"/>
              <xs:enumeration value="Coleccion"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="debeIntentarsePrevio" type="xs:boolean"/>
        <xs:element name="debeTerminarsePrevio" type="xs:boolean"/>
        <xs:element name="debeAprobarsePrevio" type="xs:boolean"/>
        <xs:element name="calificacionDeAprobacionPrevio" type="xs:double"/>
        <xs:element name="calificacionPrevioParaOmitirActual" type="xs:double"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ContextosEvaluacion.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="InterfaseAplicacion">
    <xs:sequence>
      <xs:element name="cuentaParaCalificacion" type="xs:boolean"/>
      <xs:element name="idtemEvaluacion" type="xs:integer" minOccurs="0"/>
      <xs:element name="habilitaAutoevaluacion" type="xs:boolean"/>
      <xs:element name="habilitaCoevaluacion" type="xs:boolean"/>
      <xs:element name="accesoAResolucion" type="TiposAccesoARespuesta"
default="AlumnoAccesaAlSerEnviadaActividad"/>
      <xs:element name="accesoAComentarioInstructor" type="TiposAccesoARespuesta"
default="AlumnoAccesaAlSerEnviadaActividad"/>
      <xs:element name="accesoACalificacion" type="TiposAccesoARespuesta"
default="AlumnoAccesaAlSerEnviadaActividad"/>
      <xs:element name="accesoARespuestaGuia" type="TiposAccesoARespuesta"
default="AlumnoAccesaAlSerEnviadaActividad"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="InterfaseComputo">
    <xs:sequence>
      <xs:element name="esOpcional" type="xs:boolean"/>
      <xs:element name="FormulaComputo" type="FormulasComputo" minOccurs="0"/>
      <xs:element name="valor" type="xs:double"/>
      <xs:element name="tipoValor" type="TiposValor"/>
      <xs:element name="distribucionValor" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Auto"/>
            <xs:enumeration value="Manual"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EsquemaEvaluacion">
    <xs:sequence>
      <xs:element ref="ItemEvaluacion" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ItemEvaluacion">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="idItemEvaluacion" type="xs:integer"/>
        <xs:element name="nombre" type="xs:string"/>
        <xs:element name="valor" type="xs:double"/>
        <xs:element name="tipoValor" type="TiposValor"/>
        <xs:element name="tipoEvaluacion">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Objetiva"/>
              <xs:enumeration value="Subjetiva"/>
              <xs:enumeration value="Votacion"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="TiposAccesoARespuesta">
    <xs:restriction base="xs:string">
      <xs:enumeration value="AlumnoAccesaAlSerEnviadaActividad"/>
      <xs:enumeration value="AlumnoAccesaAlSerCalificadaActividad"/>
      <xs:enumeration value="EquipoAccesaAlSerEnviadaActividad"/>
      <xs:enumeration value="EquipoAccesaAlSerCalificadaActividad"/>
      <xs:enumeration value="ClaseAccesaAlSerEnviadaActividad"/>
      <xs:enumeration value="ClaseAccesaAlSerCalificadaActividad"/>
    </xs:restriction>
  </xs:simpleType>

```

```

    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TiposValor">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Porcentaje"/>
    <xs:enumeration value="Puntuacion"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FormulasComputo">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Bruta"/>
    <xs:enumeration value="Promedio"/>
    <xs:enumeration value="DesviacionStandard"/>
    <xs:enumeration value="z"/>
    <xs:enumeration value="Percentiles"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Calificaciones.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Calificacion">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="idCalificacion" type="xs:integer"/>
        <xs:element ref="Respuesta" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="evaluacionSubjetiva" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="totalCalificacionesObjetivas" type="xs:double" default="0"/>
        <xs:element name="totalCalificacionesSubjetivas" type="xs:double" default="0"/>
        <xs:element name="calificacionTotal" type="xs:double">
          </xs:element>
        <xs:element name="puntajeExtra" type="xs:double"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Respuesta">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="idRespuesta" type="xs:integer"/>
        <xs:element name="idUnidadDeEstudio" type="xs:integer"/>
        <xs:element name="fechaRespuesta" type="xs:date"/>
        <xs:choice>
          <xs:element name="textoRespuesta" type="xs:string"/>
          <xs:element name="idObjetoAprendizajeRespuesta" type="xs:integer"
maxOccurs="unbounded"/>
        </xs:choice>
        <xs:element name="fechaRevision" type="xs:date" minOccurs="0">
          </xs:element>
        <xs:choice minOccurs="0">
          <xs:element name="textoRevision" type="xs:string"/>
          <xs:element name="idObjetoAprendizajeRevision" type="xs:integer"
maxOccurs="unbounded"/>
        </xs:choice>
        <xs:element name="calificacionInstructor" type="xs:double" default="0"/>
        <xs:element name="autoEvaluacion" type="xs:double" minOccurs="0"/>
        <xs:choice>
          <xs:element ref="Coevaluacion" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Votacion" minOccurs="0" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:element ref="RespuestaItem" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="statusRespuesta">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Enviada"/>

```

```

        <xs:enumeration value="Revisada"/>
        <xs:enumeration value="Rechazada"/>
        <xs:enumeration value="Aprobada"/>
        <xs:enumeration value="Reprogramada"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RespuestaItem">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="idRespuestaItem" type="xs:int"/>
            <xs:element name="idItemUnidadDeEstudio" type="xs:integer"/>
            <xs:choice>
                <xs:element name="textoRespuesta" type="xs:string"/>
                <xs:element name="idObjetoAprendizajeRespuesta" type="xs:integer"/>
            </xs:choice>
            <xs:element name="textoRevision" type="xs:string" minOccurs="0"/>
            <xs:element name="calificacion" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Coevaluacion">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="EvaluacionType">
                <xs:sequence>
                    <xs:element name="idCoevaluacion" type="xs:integer"/>
                    <xs:element name="calificacion" type="xs:double"/>
                    <xs:element name="comentario" type="xs:string"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="Votacion">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="EvaluacionType">
                <xs:sequence>
                    <xs:element name="idVotacion" type="xs:integer"/>
                    <xs:element name="voto" type="xs:byte"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="evaluacionSubjetiva">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="idEvaluacionSubjetiva" type="xs:integer"/>
            <xs:element name="idItemEvaluacion" type="xs:integer"/>
            <xs:element name="fechaEvaluacion" type="xs:date"/>
            <xs:element name="calificacion" type="xs:double"/>
            <xs:element name="comentario" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="EvaluacionType">
    <xs:sequence>
        <xs:element name="idUsuarioEvaluador" type="xs:integer"/>
        <xs:element name="fechaEvaluacion" type="xs:date"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

APÉNDICE B

DOCUMENTACIÓN DE LOS ESQUEMAS XML DEL SISTEMA EVALUADOR Y ESQUEMAS ACCESORIOS

Se presentan inicialmente esquemas que simulan el LMS y otros objetos, que se presupone existen, para después describir los esquemas propios de la propuesta de estándar, los cuales hacen referencia a elementos del LMS. Los primeros se indican con las siglas LMS, y los segundos con la palabra Evaluador.

Documentación del esquema Cursos (LMS)

Archivo: Cursos.xsd

Este esquema debe ser sobrescrito o extendido para incluir los elementos particulares de un LMS. Su función es servir de contenedor para todos los elementos participantes del proceso de enseñanza-aprendizaje.

Elementos:

- idCurso
- datosCurso
- instructor
- alumno
- Objetivo
- Seccion
- EsquemaEvaluacion

Curso

<p>diagrama</p>	
<p>código fuente</p>	<pre> <xs:element name="Curso"> <xs:complexType> <xs:sequence> <xs:element name="idCurso" type="xs:integer"/> <xs:element name="datosCurso"> <xs:complexType> <xs:sequence> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="tiempo" type="InterfaseTiempo"/> <xs:element name="puntajeMaximo" type="xs:integer"/> <xs:element name="puntajeMinimo" type="xs:integer"/> <xs:element name="puntajeMinimoParaAprobar" type="xs:integer"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="instructor" type="UsuarioType" maxOccurs="unbounded"/> </pre>

```

<xs:element name="alumno" minOccurs="0"
maxOccurs="unbounded">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="UsuarioType">
        <xs:sequence>
          <xs:element name="fechaIngreso"
type="xs:date"/>
          <xs:element name="fechaEgreso" type="xs:date"
minOccurs="0"/>
          <xs:element name="status">
            simpleType>
              restriction base="xs:string">
                enumeration value="Solicitando"/>
                enumeration value="Inscrito"/>
                enumeration value="Rechazado"/>
                enumeration value="Aprobado"/>
                enumeration value="Reprobado"/>
                enumeration value="Condicional"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element ref="Calificacion"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  </xs:element>
  <xs:element ref="Objetivo" maxOccurs="unbounded"/>
  <xs:element ref="Seccion" maxOccurs="unbounded"/>
  <xs:element name="EsquemaEvaluacion"
type="EsquemaEvaluacion"/>
</xs:sequence>
</xs:complexType>
</xs:element>

```

Descripción: Objeto que representa el programa de estudio o curso; desde el punto de vista del sistema es el contenedor para todos los elementos participantes del proceso de enseñanza-aprendizaje.

Cardinalidad: 1 (para el evaluador); 0-∞ (LMS).

Curso.idCurso

tipo	xs:integer
código fuente	<xs:element name="idCurso" type="xs:integer"/>

Descripción: El identificador del Curso y de todos los otros objetos que requieran uno es de tipo entero (integer). idCurso implementa cualquier valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Curso.datosCurso

diagrama	<p>The diagram illustrates the structure of the 'datosCurso' complex type. It is composed of a sequence of five elements: 'relacion' (type InterfaseRelacion), 'tiempo' (type InterfaseTiempo), 'puntajeMaximo' (type xs:integer), 'puntajeMinimo' (type xs:integer), and 'puntajeMinimoParaAprobar' (type xs:integer). Each element is represented by a box with its name and type, and a '+' sign indicating a cardinality of 1. The elements are connected to a central 'datosCurso' box via a sequence container.</p>
código fuente	<pre><xs:element name="datosCurso"> <xs:complexType> <xs:sequence> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="tiempo" type="InterfaseTiempo"/> <xs:element name="puntajeMaximo" type="xs:integer"/> <xs:element name="puntajeMinimo" type="xs:integer"/> <xs:element name="puntajeMinimoParaAprobar" type="xs:integer"/> </xs:sequence> </xs:complexType> </xs:element></pre>

Descripción: datosCurso implementa el contexto de uso de un curso, que puede sobrescribirse para expresar los datos generales de un curso (por ejemplo, el nombre, los créditos, el grado escolar, etc.), siempre que se conserve dicho contexto de uso; en este caso se ha hecho implementando las interfases de relación y de tiempo. Como un ensayo se incluyen datos adicionales, los relativos al puntaje.

Cardinalidad: 1.

Curso.datosCurso.relacion

tipo	InterfaseRelacion
código fuente	<xs:element name="relacion" type="InterfaseRelacion"/>

Descripción: Implementación de InterfaseRelacion (ver más adelante), que forma parte del contexto de uso, junto con InterfaseTiempo.

Cardinalidad: 1.

Curso.datosCurso.tiempo

tipo	InterfaseTiempo
código fuente	<xs:element name="tiempo" type="InterfaseTiempo"/>

Descripción: Implementación de InterfaseTiempo (ver más adelante), que forma parte del contexto de uso, junto con InterfaseRelacion.

Cardinalidad: 1.

Curso.datosCurso.puntajeMaximo

Tipo	xs:integer
código fuente	<xs:element name="puntajeMaximo" type="xs:integer"/>

Descripción: puntajeMaximo se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobrescriba este esquema con el de un LMS en particular. puntajeMaximo expresa el número de puntos que vale el curso cuando es evaluable (una cifra común es 100).

Cardinalidad: 1.

Curso.datosCurso.puntajeMinimo

tipo	xs:integer
código fuente	<xs:element name="puntajeMinimo" type="xs:integer"/>

Descripción: puntajeMinimo se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobrescriba este esquema con el de un LMS en particular. puntajeMinimo expresa el número de puntos que pueden obtenerse cuando el curso es evaluable; podría ser 0 o más, como ciertos cursos de capacitación que por el hecho de inscribirse, independientemente del desempeño, ya asignan un crédito o puntos.

Cardinalidad: 1.

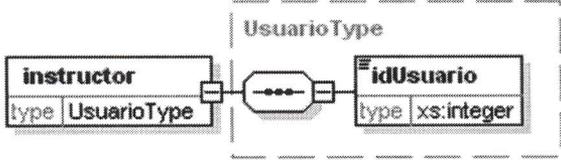
Curso.datosCurso.puntajeMinimoParaAprobar

tipo	xs:integer
código fuente	<xs:element name="puntajeMinimoParaAprobar" type="xs:integer"/>

Descripción: `puntajeMinimoParaAprobar` se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobrescriba este esquema con el de un LMS en particular. `puntajeMinimoParaAprobar` expresa el número mínimo de puntos que se requieren para acreditar o aprobar el curso cuando es evaluable.

Cardinalidad: 1.

Curso.instructor

diagrama	
tipo	UsuarioType
código fuente	<pre><xs:element name="instructor" type="UsuarioType" maxOccurs="unbounded"/></pre>

Descripción: `Instructor` es un objeto de tipo `UsuarioType` (ver más adelante). Para el sistema evaluador basta con referenciarlo con su identificador. Puede sobrescribirse para incluir otros detalles necesarios para el LMS.

Cardinalidad: 1-∞.

Curso.alumno

<p>diagrama</p>	<p>The diagram illustrates an XSD extension. A box labeled 'alumno' (type UsuarioType) is connected via a dashed line to a dashed box labeled 'UsuarioType'. Inside 'UsuarioType', there is an element 'idUsuario' (type xs:integer). Below 'UsuarioType', there is a sequence of elements: 'fechaIngreso' (type xs:date), 'fechaEgreso' (type xs:date), 'status' (type xs:string), and 'Calificacion' (type). The 'fechaEgreso' element is shown with a dashed border, indicating it is not present in the base type. The 'alumno' element is connected to the 'idUsuario' element and the sequence of elements.</p>
<p>tipo</p>	<p>Extensión de UsuarioType</p>
<p>código fuente</p>	<pre> <xs:element name="alumno" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:complexContent> <xs:extension base="UsuarioType"> <xs:sequence> <xs:element name="fechaIngreso" type="xs:date"/> <xs:element name="fechaEgreso" type="xs:date" minOccurs="0"/> <xs:element name="status"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Solicitando"/> <xs:enumeration value="Inscrito"/> <xs:enumeration value="Rechazado"/> <xs:enumeration value="Aprobado"/> <xs:enumeration value="Reprobado"/> <xs:enumeration value="Condicional"/> </xs:restriction> </xs:simpleType> </xs:element> <xs:element ref="Calificacion"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

	<pre></xs:complexType> </xs:element></pre>
--	--

Descripción: alumno es una extensión de UsuarioType (ver más adelante); puede sobreescribirse para contener los datos relevantes de los usuarios de un LMS particular, con tal de que cada objeto alumno conserve un identificador único. En este caso se ensaya un grupo de elementos al extender UsuarioType: las fechas de ingreso y egreso, el status, en la forma de una enumeración de todos los status válidos, y un campo de calificación.

Cardinalidad: 0-∞.

Curso.alumno.fechaIngreso

tipo	xs:date
código fuente	<pre><xs:element name="fechaIngreso" type="xs:date"/></pre>

Descripción: fechaIngreso se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobreescriba este esquema con el de un LMS en particular. fechaIngreso puede expresar tanto la fecha en que el alumno solicita ingresar al curso, como la fecha en que recibe respuesta a su solicitud.

Cardinalidad: 1.

Curso.alumno.fechaEgreso

tipo	xs:date
código fuente	<pre><xs:element name="fechaEgreso" type="xs:date" minOccurs="0"/></pre>

Descripción: fechaEgreso se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobrescriba este esquema con el de un LMS en particular.

fechaEgreso expresa la fecha en que el alumno deja de ser parte del curso.

Cardinalidad: 0-1 (el dato existe hasta que ocurre el egreso).

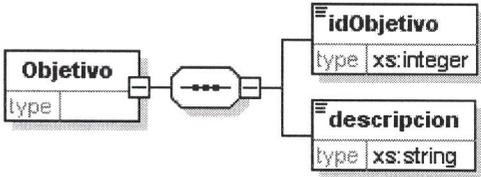
Curso.alumno.status

tipo	Restricción de xs:string
facets	enumeration Solicitando enumeration Inscrito enumeration Rechazado enumeration Aprobado enumeration Reprobado enumeration Condicional
código fuente	<pre> <xs:element name="status"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Solicitando"/> <xs:enumeration value="Inscrito"/> <xs:enumeration value="Rechazado"/> <xs:enumeration value="Aprobado"/> <xs:enumeration value="Reprobado"/> <xs:enumeration value="Condicional"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

Descripción: status se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobrescriba este esquema con el de un LMS en particular. Aquí status es una enumeración de las diferentes relaciones que puede tener un alumno con un curso: solicitando, inscrito, rechazado, aprobado, reprobado, condicional.

Cardinalidad: 1.

Curso.Objetivo

diagrama	
código fuente	<pre><xs:element name="Objetivo"> <xs:complexType> <xs:sequence> <xs:element name="idObjetivo" type="xs:integer"/> <xs:element name="descripcion" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element></pre>

Descripción: Objetivo es un elemento requerido por el evaluador. Puede sobrescribirse con tal de que conserve un identificador único.

Cardinalidad: 1-∞.

Curso.Objetivo.idObjetivo

tipo	xs:integer
código fuente	<xs:element name="idObjetivo" type="xs:integer"/>

Descripción: El identificador de Objetivo es de tipo entero (integer) y representa cualquier valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Curso.Objetivo.descripcion

tipo	xs:string
código fuente	<xs:element name="descripcion" type="xs:string"/>

Descripción: descripcion se ha incluido para ensayar alguno de los datos que puede tener el curso cuando se sobrescriba este esquema con el de un LMS en particular. Aquí descripcion es un texto que es el objetivo en sí.

Cardinalidad: 1.

Curso.Seccion

<p>diagrama</p>	
<p>tipo</p>	<p>Extensión de IEstructura</p>
<p>código fuente</p>	<pre><xs:element name="Seccion"> <xs:complexType> <xs:sequence> <xs:element name="idSeccion" type="xs:integer"/> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="tiempo" type="InterfaseTiempo"/> <xs:element ref="UnidadDeEstudio" maxOccurs="unbounded"/> <xs:element ref="Seccion" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>

Descripción: Seccion es una estructura que puede sobrescribirse para expresar los elementos organizadores del curso.

Cardinalidad: 0-∞.

Curso.Seccion.idSeccion

tipo	xs:integer
código fuente	<code><xs:element name="idSeccion" type="xs:integer"/></code>

Descripción: El identificador de Seccion es de tipo entero (integer) y representa cualquier valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Curso.Seccion.relacion

tipo	InterfaseRelacion
código fuente	<code><xs:element name="relacion" type="InterfaseRelacion"/></code>

Descripción: Implementación de InterfaseRelacion (ver más adelante), que forma parte del contexto de uso.

Cardinalidad: 1.

Curso.Seccion.tiempo

tipo	InterfaseTiempo
código fuente	<code><xs:element name="tiempo" type="InterfaseTiempo"/></code>

Descripción: Implementación de InterfaseTiempo (ver más adelante), que forma parte del contexto de uso.

Cardinalidad: 1.

Curso.Seccion.Seccion

tipo	Extensión de IEstructura
código fuente	<code><xs:element ref="Seccion" minOccurs="0" maxOccurs="unbounded"/></code>

Descripción: Objeto Seccion, idéntico al descrito anteriormente, anidado para formar subsecciones.

Cardinalidad: 0-∞.

Curso.Seccion.UnidadDeEstudio

código fuente	<code><xs:element ref="UnidadDeEstudio" maxOccurs="unbounded"/></code>
----------------------	--

Descripción: Cada objeto UnidadDeEstudio (ver más adelante) representa los contenidos del curso en conjunción con ciertos objetivos, en un contexto específico. Un esquema particular lo detalla más adelante.

Cardinalidad: 1-∞.

Curso.EschemaEvaluacion

código fuente	<code><xs:element name="EsquemaEvaluacion" type="EsquemaEvaluacion"/></code>
----------------------	--

Descripción: Este objeto es una agrupación virtual de los ítemes de evaluación que se describen más adelante.

Cardinalidad: 1.

Documentación del esquema Interfases (LMS)

Archivo: Interfases.xsd

Se simula este esquema como el repositorio de las interfases del sistema, que debe aportar el LMS. pero puede sobrescribirse. En este caso contiene tan sólo el tipo complejo IEstructura. Si este esquema se eliminara o no se incluyera IEstructura, deberán sobrescribirse los esquemas que lo implementan.

Elementos:

- IEstructura

IEstructura

<p>diagrama</p>	
<p>código fuente</p>	<pre> <xs:complexType name="IEstructura"> <xs:sequence> <xs:element name="idEstructuraSuperior" type="xs:integer" minOccurs="0"/> <xs:element name="orden"> <xs:simpleType> <xs:restriction base="xs:integer"> <xs:minInclusive value="1"/> </xs:restriction> </xs:simpleType> </xs:element> <xs:element name="nombre" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:complexType> </pre>

Descripción: IEstructura es la interfase para implementar objetos organizadores (secciones, grupos, colecciones, etc.) y para implementar los datos de ubicación que utiliza el contexto de uso.

IEstructura.idEstructuraSuperior

tipo	xs:integer
código fuente	<pre><xs:element name="idEstructuraSuperior" type="xs:integer" minOccurs="0"/></pre>

Descripción: idEstructuraSuperior es una referencia al objeto padre que contiene al que implementa esta interfase, ambos del mismo tipos. El valor puede ser cero, o nulo, ya que un objeto podría estar directamente contenido en otro elemento diferente, por ejemplo, cuando una sección pertenece al curso, idEstructuraSuperior sería 0; cuando una es subsección de otra, idEstructuraSuperior sería mayor de 0 o no nula.

Cardinalidad: 0-1

IEstructura.orden

tipo	Restricción de xs:integer
facets	minInclusive 1
código fuente	<pre><xs:element name="orden"> <xs:simpleType> <xs:restriction base="xs:integer"> <xs:minInclusive value="1"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

Descripción: orden es un número entero que permite organizar las estructuras en secuencias lineales, por lo tanto se define con valores de 1 en adelante. Puede sobrescribirse si a un LMS, aun teniendo estructuras, no le es necesario ordenar los elementos del curso.

Cardinalidad: 1

IEstructura.nombre

tipo	xs:string
código fuente	<code><xs:element name="nombre" type="xs:string" minOccurs="0"/></code>

Descripción: nombre se ha incluido para ensayar alguno de los datos que puede tener una estructura cuando se sobrescriba este esquema con el de un LMS en particular. Aquí nombre es un texto opcional que describe o sirve de título para la estructura.

Cardinalidad: 0-1

Documentación del esquema Usuarios (LMS)

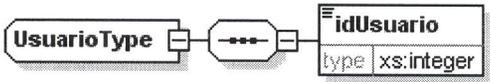
Archivo: Usuarios.xsd

Usuarios es una clase base para todos los usuarios del sistema, que puede sobrescribirse para incluir otros datos pertinentes (nombre, dirección, escolaridad, roles, etc.), con tal de que conserve un identificador único.

Elementos:

- UsuarioType

UsuarioType

diagrama	
código fuente	<pre><xs:complexType name="UsuarioType"> <xs:sequence> <xs:element name="idUsuario" type="xs:integer"/> </xs:sequence> </xs:complexType></pre>

Descripción: Define todas las clases de usuario. En el contexto del evaluador, sólo se requiere tener una referencia a él, independientemente de sus demás atributos.

UsuarioType.idUsuario

tipo	xs:integer
código fuente	<xs:element name="idUsuario" type="xs:integer"/>

Descripción: El identificador de Usuario es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Documentación del esquema ObjetosAprendizaje (LMS)

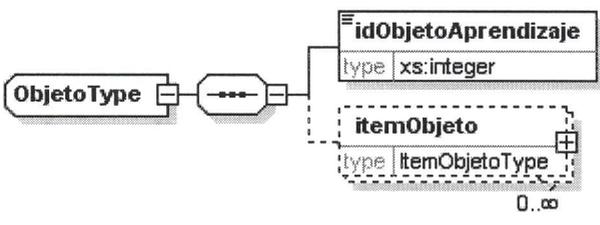
Archivo: ObjetosAprendizaje.xsd

Los objetos de aprendizaje pueden ser creados y administrador dentro del sistema o fuera de él, ya que se trata de contenidos autodefinidos y reutilizables; por ello se ha simulado un esquema separado para los objetos de aprendizaje. Además, dado que existen ya especificaciones para definirlos y manejarlos, aquí sólo se ensayan los elementos mínimos que requeriría el evaluador para funcionar.

Elementos:

- ObjetoType
- ItemObjetoType

ObjetoType

Diagrama	
código fuente	<pre><xs:complexType name="ObjetoType"> <xs:sequence> <xs:element name="idObjetoAprendizaje" type="xs:integer"/> <xs:element name="itemObjeto" type="ItemObjetoType" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType></pre>

Descripción: Describe un contenido autodefinido y reutilizable. Aunque puede ser muy complejo, para un sistema evaluador sólo son significativos su identificador y los de sus componentes, si los hay.

ObjetoType.idObjetoAprendizaje

tipo	xs:integer
código fuente	<xs:element name="idObjetoAprendizaje" type="xs:integer"/>

Descripción: El identificador de ObjetoType es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

ObjetoType.itemObjeto

tipo	ItemObjetoType
código fuente	<xs:element name="itemObjeto" type="ItemObjetoType"

minOccurs="0" maxOccurs="unbounded"/>

Descripción: itemObjeto es un contenido componente del objeto de aprendizaje; puede ser o no un objeto de aprendizaje en sí. Aquí se esquematiza como si lo fuera, definiéndolo como de tipo ItemObjetoType.

Cardinalidad: 1-∞.

ItemObjetoType

diagrama	<p>El diagrama muestra un tipo complejo 'ItemObjetoType' que contiene una secuencia de elementos. El elemento principal es 'idItemObjetoAprendizaje', que es de tipo 'xs:integer'.</p>
código fuente	<pre><xs:complexType name="ItemObjetoType"> <xs:sequence> <xs:element name="idItemObjetoAprendizaje" type="xs:integer"/> </xs:sequence> </xs:complexType></pre>

Descripción: Tal como en el caso del ObjetoType, sus componentes secundarios, los ItemObjetoType puede poseer muchos elementos y ser elementos complejos. Sin embargo, aquí se ensayan las características esenciales.

ItemObjetoType.idItemObjetoAprendizaje

Tipo	xs:integer
código fuente	<xs:element name="idItemObjetoAprendizaje" type="xs:integer"/>

Descripción: El identificador de este objeto es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Documentación del esquema UnidadesEstudio (LMS-Evaluador)

Archivo: UnidadesEstudio.xsd

En esta estructura pueden encontrarse el sistema LMS y el evaluador; esto ocurre cuando a una unidad de estudio se le agrega un contexto de evaluación; en otras palabras, cuando se programa que un contenido del curso sea evaluable. Como se verá más adelante, precisamente el contexto de evaluación, las interfases de aplicación y cómputo, se presentan como opcionales. Ambas interfases pertenecen a la propuesta de estándar, mientras que el resto de la descripción de la unidad de estudio es una simulación de los elementos que podría contener, a grandes rasgos, en un LMS en general.

Elementos:

- idUnidadDeEstudio
- idObjetivo
- idObjetoAprendizaje
- Relacion
- Tiempo
- AplicacionEvaluacion
- Computo
- SeccionUnidad

UnidadDeEstudio

<p>diagrama</p>	
<p>código fuente</p>	<pre> <xs:element name="UnidadDeEstudio"> <xs:complexType> <xs:sequence> <xs:element name="idUnidadDeEstudio" type="xs:integer"/> <xs:element name="idObjetivo" type="xs:integer" maxOccurs="unbounded"/> <xs:element name="idObjetoAprendizaje" type="xs:integer"/> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="tiempo" type="InterfaseTiempo"/> <xs:element name="aplicacionEvaluacion" type="InterfaseAplicacion" minOccurs="0"/> <xs:element name="computo" type="InterfaseComputo" minOccurs="0"/> <xs:element name="seccionUnidad" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="idSeccionUnidad" </pre>

	<pre> type="xs:integer"/> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="computo" type="InterfaseComputo"/> <xs:element name="itemUnidad" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> element name="idItemUnidad" type="xs:integer"/> element name="idItemObjetoAprendizaje" type="xs:integer"/> element name="relacion" type="InterfaseRelacion"/> element name="computo" type="InterfaseComputo"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> <xs:element ref="UnidadDeEstudio" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>
--	---

Descripción: La unidad de estudio puede tratarse tanto de un contenido genérico como de una actividad de evaluación.

UnidadDeEstudio.idUnidadEstudio

tipo	xs:integer
código fuente	<xs:element name="idUnidadEstudio" type="xs:integer"/>

Descripción: El identificador de UnidadDeEstudio es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

UnidadDeEstudio.idObjetivo

código fuente	<code><xs:element name="idObjetivo" type="xs:integer" minOccurs="unbounded"/></code>
----------------------	--

Descripción: Objetivo es un elemento requerido para definir por completo una unidad de estudio anidada, ya que ésta, pedagógicamente hablando, debe cumplir uno o más objetivos de aprendizaje del curso (véase el elemento objetivo en el esquema correspondiente al curso). Aquí como referencia el valor del identificador del objetivo correspondiente, que ya debió haber sido definido entre los datos del curso.

Cardinalidad: 1-∞.

UnidadDeEstudio.idObjetoAprendizaje

código fuente	<code><xs:element name="idObjetoAprendizaje" type="xs:integer"/></code>
----------------------	---

Descripción: Referencia al objeto de aprendizaje (véase antes la descripción del esquema correspondiente) que forma la unidad de estudio.

Cardinalidad: 1.

UnidadDeEstudio.relacion

tipo	InterfaseRelacion
código fuente	<code><xs:element name="relacion" type="InterfaseRelacion"/></code>

Descripción: Implementación de InterfaseRelacion (véase más adelante), que forma parte del contexto de uso, junto con InterfaseTiempo.

Cardinalidad: 1.

UnidadDeEstudio.tiempo

tipo	InterfaseTiempo
código fuente	<code><xs:element name="tiempo" type="InterfaseTiempo"/></code>

Descripción: Implementación de InterfaseTiempo (véase más adelante), que forma parte del contexto de uso, junto con InterfaseRelacion.

Cardinalidad: 1.

UnidadDeEstudio.aplicacionEvaluacion

tipo	InterfaseAplicacion
código fuente	<code><xs:element name="aplicacionEvaluacion" type="InterfaseAplicacion" minOccurs="0"/></code>

Descripción: Implementación de InterfaseAplicacion (véase más adelante), que forma parte del contexto de evaluación, junto con InterfaseComputo. Se trata de una implementación opcional; al utilizarse, se transforma la unidad de estudio en una actividad de evaluación.

Cardinalidad: 0-1.

UnidadDeEstudio.computo

tipo	InterfaseComputo
código fuente	<code><xs:element name="computo" type="InterfaseComputo" minOccurs="0"/></code>

Descripción: Implementación de InterfaseComputo (véase más adelante), que forma parte del contexto de evaluación, junto con InterfaseAplicacion. Se trata de una implementación opcional; al utilizarse, se transforma la unidad de estudio en una actividad de evaluación.

Cardinalidad: 0-1.

UnidadDeEstudio.seccionUnidad

<p>diagrama</p>	
<p>código fuente</p>	<pre> <xs:element name="seccionUnidad" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="idSeccionUnidad" type="xs:integer"/> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="computo" type="InterfaseComputo"/> <xs:element name="itemUnidad" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="idItemUnidad" type="xs:integer"/> <xs:element name="idItemObjetoAprendizaje" type="xs:integer"/> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="computo" type="InterfaseComputo"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </pre>

Descripción: seccionUnidad es una estructura que organiza ítemes componentes de una unidad de estudio, cuando tales ítemes existen. El diagrama de UnidadDeEstudio revela que una unidad puede componerse de otras unidades; pero se ha elegido no utilizar las estructuras Seccion (que sí se usan en el curso en general), sino seccionUnidad para mostrar que los

ítemes no son necesariamente unidades de estudio, es decir, elementos independientes, y que están agrupados en un solo nivel, ya que no hay subsecciones de las secciones de la unidad. Con esta estructura se describen contenidos como los exámenes o los webquest. En cambio las unidades de estudio anidadas son independientes de la unidad de estudio superior y pueden verse más bien como anexos o extensiones, más que como ítemes componentes. Un ejemplo sería un contenido conformado por un texto, cuyas subunidades de estudio son imágenes, archivos y url anexos.

Cardinalidad: 0-∞.

UnidadDeEstudio.seccionUnidad.idSeccionUnidad

tipo	xs:integer
código fuente	<xs:element name="idSeccionUnidad" type="xs:integer"/>

Descripción: El identificador de una sección de unidad es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

UnidadDeEstudio.seccionUnidad.relacion

tipo	InterfaseRelacion
código fuente	<xs:element name="relacion" type="InterfaseRelacion"/>

Descripción: Implementación de InterfaseRelacion, que forma parte del contexto de uso, junto con InterfaseTiempo. Esta última no se aplica aquí ya que, por tratarse de un elemento dependiente de la unidad, se administra en el mismo contexto de tiempo que ella.

Cardinalidad: 1.

UnidadDeEstudio.seccionUnidad.computo

tipo	InterfaseComputo
código fuente	<xs:element name="computo" type="InterfaseComputo"/>

Descripción: Implementación de InterfaseComputo, que forma parte del contexto de evaluación, junto con InterfaseAplicación. Esta última no se aplica aquí ya que, por tratarse de un elemento dependiente de la unidad, se administra en el mismo contexto de aplicación que ella. Es, además, un dato requerido porque se considera que una unidad de estudio sólo tiene secciones e ítems cuando se trata de una actividad de evaluación.

Cardinalidad: 1.

UnidadDeEstudio.seccionUnidad.itemUnidad

diagrama	<p>The diagram illustrates the structure of the 'itemUnidad' element. It is a sequence container (represented by a rectangle with a small square on the left) that contains a sequence (represented by a rounded rectangle with three dots) of four elements. The elements are: 'idItemUnidad' (type xs:integer), 'idItemObjetoAprendizaje' (type xs:integer), 'relacion' (type InterfaseRelacion), and 'computo' (type InterfaseComputo). Each element is shown in a box with its name and type, and the 'relacion' and 'computo' elements have a '+' sign indicating they are optional.</p>
código fuente	<pre><xs:element name="itemUnidad" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="idItemUnidad" type="xs:integer"/> <xs:element name="idItemObjetoAprendizaje" type="xs:integer"/> <xs:element name="relacion" type="InterfaseRelacion"/> <xs:element name="computo" type="InterfaseComputo"/> </xs:sequence> </xs:complexType> </xs:element></pre>

Descripción: itemUnidad representa a los objetos que pueden componer una actividad de evaluación, como las preguntas de un examen, que, aunque pueden crearse independientes, en su uso sólo tienen sentido cuando forman parte de un examen.

Cardinalidad: 1-∞.

UnidadDeEstudio.seccionUnidad.itemUnidad.idItemUnidad

tipo	xs:integer
código fuente	<xs:element name="idItemUnidad" type="xs:integer"/>

Descripción: El identificador de cada ítem es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

UnidadDeEstudio.seccionUnidad.itemUnidad.idItemObjetoAprendizaje

tipo	ItemObjetoType
código fuente	<xs:element name=" idItemObjetoAprendizaje " type="ItemObjetoType"/>

Descripción: Este es el ítem del objeto de aprendizaje con el que se corresponde el ítem de la unidad. Recuérdese que un elemento constituyente de una unidad de estudio es un objeto de aprendizaje; si éste tiene componentes, la unidad de estudio debe contar con un mecanismo para referenciarlos.

Cardinalidad: 1.

UnidadDeEstudio.seccionUnidad.itemUnidad.relacion

tipo	InterfaseRelacion
código fuente	<xs:element name="relacion" type="InterfaseRelacion"/>

Descripción: Implementación de InterfaseRelacion, que forma parte del contexto de uso, junto con InterfaseTiempo. Esta última no se aplica aquí ya que, por tratarse de un elemento dependiente de la unidad, se administra en el mismo contexto de tiempo que ella.

Cardinalidad: 1.

UnidadDeEstudio.seccionUnidad.itemUnidad.computo

tipo	InterfaseComputo
código fuente	<xs:element name="computo" type="InterfaseComputo"/>

Descripción: Implementación de InterfaseComputo, que forma parte del contexto de evaluación, junto con InterfaseAplicación. Esta última no se aplica aquí ya que, por tratarse de un elemento dependiente de la unidad, se administra en el mismo contexto de aplicación que ella. Es, además, un dato requerido porque se considera que una unidad de estudio sólo tiene secciones e ítemes cuando se trata de una actividad de evaluación.

Cardinalidad: 1.

UnidadDeEstudio.UnidadDeEstudio

código fuente	<xs:element ref="UnidadDeEstudio" minOccurs="0" maxOccurs="unbounded"/>
----------------------	---

Descripción: Unidad de estudio anidada, idéntica al objeto ya descrito (ver detalles arriba). Este anidamiento describe más bien una suerte de anexión o extensión de contenidos, más que una subordinación.

Cardinalidad: 0-∞.

Documentación del esquema ContextosUso (LMS)

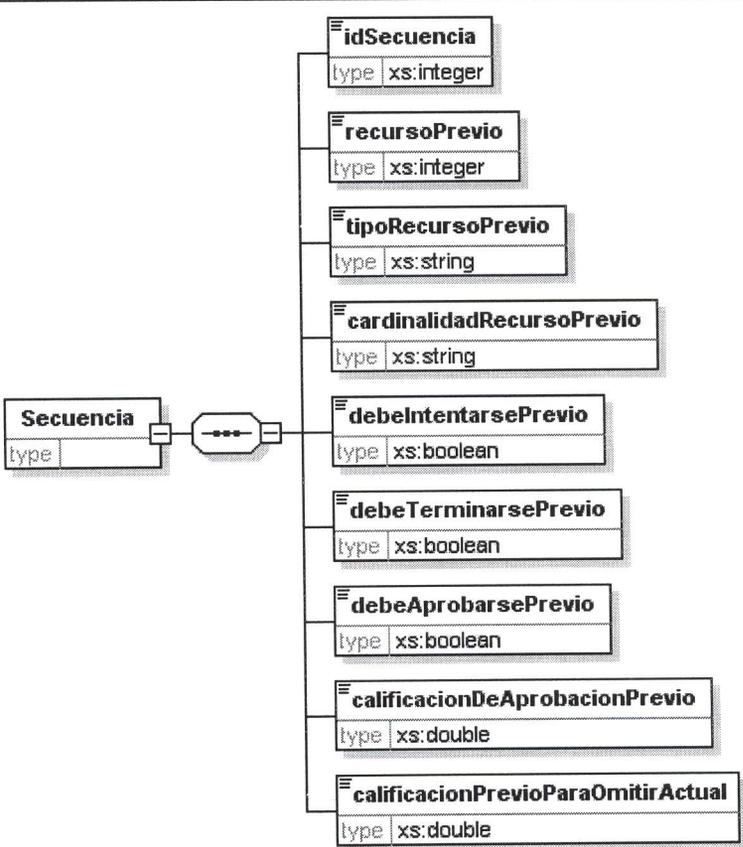
Archivo: ContextoUso.xsd

Este esquema simula el contexto de uso de un curso y de sus componentes. Se han incluido los elementos básicos, por lo que sólo sería necesario extenderlos. Sin embargo, si un LMS prescinde de las interfases y el secuenciador que componen este contexto de uso, deberán modificarse los esquemas del evaluador que los implementan.

Elementos:

- Secuencia
- InterfaseRelacion
- InterfaseTiempo

Secuencia

<p>diagrama</p> 	
<p>código fuente</p>	<pre><xs:element name="Secuencia"> <xs:complexType> <xs:sequence></pre>

```

<xs:element name="idSecuencia" type="xs:integer"/>
<xs:element name="recursoPrevio" type="xs:integer"/>
<xs:element name="tipoRecursoPrevio">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Curso"/>
      <xs:enumeration value="Seccion"/>
      <xs:enumeration value="Unidad"/>
      <xs:enumeration value="SeccionUnidad"/>
      <xs:enumeration value="ItemUnidad"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="cardinalidadRecursoPrevio">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Item"/>
      <xs:enumeration value="Coleccion"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="debeIntentarsePrevio"
type="xs:boolean"/>
<xs:element name="debeTerminarsePrevio"
type="xs:boolean"/>
<xs:element name="debeAprobarsePrevio"
type="xs:boolean"/>
<xs:element name="calificacionDeAprobacionPrevio"
type="xs:double"/>
<xs:element name="calificacionPrevioParaOmitirActual"
type="xs:double"/>
</xs:sequence>
</xs:complexType>
</xs:element>

```

Descripción: Una secuencia define las condiciones que deben buscarse en un recurso previo (un *item* o *coleccion*) para que un recurso actual pueda tener algún comportamiento predefinido por el sistema. Esta simulación muestra algunas posibles condiciones. Este desarrollo puede sobrescribirse con elementos particulares del LMS o con estándares existentes para el caso.

Secuencia.idSecuencia

tipo	xs:integer
código fuente	<code><xs:element name="idSecuencia" type="xs:integer"/></code>

Descripción: El identificador de la secuencia es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Secuencia.recursoPrevio

tipo	xs:integer
código fuente	<code><xs:element name="recursoPrevio" type="xs:integer"/></code>

Descripción: recursoPrevio es de tipo entero (integer) y representa una referencia al identificador (ID) de un objeto externo a éste. Este objeto puede ser del mismo tipo que el objeto que implementa la secuencia, o puede tratarse de una colección de objetos. Por ejemplo, el tema Y implementa una secuencia que define como recurso previo al tema X; para acceder al tema Y, el alumno debe resolver las actividades del tema X.

Cardinalidad: 1.

Secuencia.tipoRecursoPrevio

tipo	Restricción de xs:string
facets	enumeration Curso enumeration Seccion enumeration Unidad enumeration SeccionUnidad enumeration ItemUnidad
código fuente	<code><xs:element name="tipoRecursoPrevio"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Curso"/> <xs:enumeration value="Seccion"/></code>

	<pre> <xs:enumeration value="Unidad"/> <xs:enumeration value="SeccionUnidad"/> <xs:enumeration value="ItemUnidad"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>
--	---

Descripción: tipoRecursoPrevio es una enumeración para definir a recursoPrevio; en realidad, según aquí propone, la dupla recursoPrevio-tipoRecursoPrevio son el identificador del recurso que antecede al implementador de la secuencia. Los objetos que pueden ser tal recurso previo se ofrecen en orden de general a particular: Curso, Seccion, Unidad, SeccionUnidad, ItemUnidad. La enumeración puede extenderse con otros valores válidos para el sistema.

Cardinalidad: 1.

Secuencia.cardinalidadRecursoPrevio

tipo	restricción de xs:string
facets	enumeration Item enumeration Coleccion
código fuente	<pre> <xs:element name="cardinalidadRecursoPrevio"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Item"/> <xs:enumeration value="Coleccion"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

Descripción: Esta enumeración facilita definir si el recurso previo es un objeto individual o una colección de objetos. La enumeración puede extenderse con otros valores válidos para el sistema.

Cardinalidad: 1.

Secuencia.debeIntentarsePrevio

tipo	xs:boolean
código fuente	<xs:element name="debeIntentarsePrevio" type="xs:boolean"/>

Descripción: Bandera booleana que define si es obligatorio intentar el recurso previo o no para continuar con el recurso actual (el que implementa la secuencia). El LMS debe definir el significado preciso de intentar; por ejemplo, podría ser acceder un contenido, o bien, al menos comenzar a resolver una actividad.

Cardinalidad: 1.

Secuencia.debeTerminarsePrevio

tipo	xs:boolean
código fuente	<xs:element name="debeTerminarsePrevio" type="xs:boolean"/>

Descripción: Bandera booleana que implica la anterior y define si es obligatorio terminar por completo el recurso previo o no para continuar con el recurso actual (el que implementa la secuencia). El LMS debe definir el significado preciso de terminar; por ejemplo, podría ser acceder todo un contenido y sus componentes, o bien, resolver toda una actividad y enviar la resolución.

Cardinalidad: 1.

Secuencia.debeAprobarsePrevio

tipo	xs:boolean
código fuente	<xs:element name="debeAprobarsePrevio" type="xs:boolean"/>

Descripción: Bandera booleana que define si es obligatorio obtener una calificación aprobatoria en el recurso previo o no para continuar con el recurso actual (el que implementa la secuencia). El LMS debe definir el significado preciso de aprobar; por ejemplo, podría ser acceder una sección del curso y obtener en promedio notas positivas en las actividades de evaluación que la componen.

Cardinalidad: 1.

Secuencia.calificacionDeAprobacionPrevio

tipo	xs:double
código fuente	<xs:element name="calificacionDeAprobacionPrevio" type="xs:double"/>

Descripción: Propiedad que complementa la anterior, para definir cuál es la calificación mínima de aprobación que debe considerarse para el recurso previo. Si debeAprobarsePrevio es falso, entonces, calificacionDeAprobacionPrevio carece de significado; con todo, se recomienda como campo requerido.

Cardinalidad: 1.

Secuencia.calificacionPrevioParaOmitirActual

tipo	xs:double
código fuente	<xs:element name="calificacionPrevioParaOmitirActual" type="xs:double"/>

Descripción: Esta propiedad indica la calificación que, de alcanzarse, permitiría al alumno pasar por alto el recurso actual (el que implementa la secuencia). El LMS debe definir cómo computaría este último, si el alumno optara por omitirlo; una solución sería que, si el alumno exenta una actividad, reciba como nota, el promedio que le permitió exentar.

Cardinalidad: 1.

InterfaseRelacion

diagrama	
código fuente	<pre><xs:complexType name="InterfaseRelacion"> <xs:sequence> <xs:element name="ubicacion" type="IEstructura"/> <xs:element ref="Secuencia"/> </xs:sequence> </xs:complexType></pre>

Descripción: Interfase que define la ubicación de un elemento del curso (curso, sección, unidad, sección de unidad o ítem de unidad) y el comportamiento que tiene en la secuencia de los otros elementos con los que se relaciona. Para ello implementa IEstructura y Secuencia, explicados antes. InterfaseRelacion puede sobrescribirse con tal de que el LMS ofrezca al evaluador una estructura de datos o funcionalidad equivalentes.

InterfaseRelacion.ubicación

tipo	IEstructura
código fuente	<xs:element name="ubicacion" type="IEstructura"/>

Descripción: Implementación de IEstructura que hace esta interfase para expresar la ubicación del objeto que la implementa.

Cardinalidad: 1

InterfaseRelacion.Secuencia

tipo	IEstructura
-------------	-------------

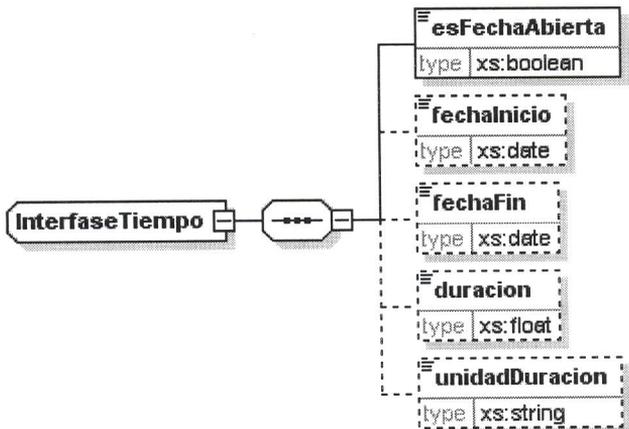
código fuente

```
<xs:element ref="Secuencia"/>
```

Descripción: Implementación de Secuencia que hace esta interfase para expresar su comportamiento en relación con otros elementos del curso. Aunque aquí se presenta como un elemento requerido para la interfase, en realidad puede considerarse opcional; se ha esta cardinalidad para subrayar que, incluso si no se define una secuencia, el LMS debe tener implícita alguna.

Cardinalidad: 1

InterfaseTiempo

<p>diagrama</p>	
<p>código fuente</p>	<pre><xs:complexType name="InterfaseTiempo"> <xs:sequence> <xs:element name="esFechaAbierta" type="xs:boolean"/> <xs:element name="fechaInicio" type="xs:date" minOccurs="0"/> <xs:element name="fechaFin" minOccurs="0"> <xs:simpleType> <xs:restriction base="xs:date"/> </xs:simpleType> </xs:element> <xs:element name="duracion" type="xs:float" minOccurs="0"/> <xs:element name="unidadDuracion" minOccurs="0"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Mes"/> </xs:restriction> </xs:simpleType> </xs:element> </xs:sequence> </xs:complexType></pre>

```

        <xs:enumeration value="Dia"/>
        <xs:enumeration value="Hora"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

Descripción: Interfase que define el marco de tiempo en el que se desarrolla el objeto que la implementa; idealmente el LMS debe ofrecer un mecanismo para controlar que los elementos del curso tengan un contexto de uso compatible con el contexto del curso en general; por ejemplo, no se esperaría que una unidad de estudio esté disponible en una fecha cuando el curso en general no lo está. Esta interfase puede sobrescribirse, siempre que el LMS provea al evaluador una estructura de datos o funcionalidad equivalentes.

InterfaseTiempo.esFechaAbierta

tipo	xs:boolean
código fuente	<xs:element name="esFechaAbierta" type="xs:boolean"/>

Descripción: Bandera booleana para indicar si hay restricciones de fecha para el objeto que implementa esta interfase; si el valor es falso, entonces fechaInicio es requerido; de lo contrario, carece de significado.

Cardinalidad: 1.

InterfaseTiempo.fechaInicio

tipo	xs:date
código fuente	<xs:element name="fechaInicio" type="xs:date" minOccurs="0"/>

Descripción: Indica la fecha de apertura de un objeto; el significado de dicha apertura depende del LMS, por ejemplo, puede ser la fecha en que los alumnos tienen acceso, o la fecha a partir de la cual una actividad puede ser resuelta, etc. Este dato es requerido cuando esFechaAbierta tiene valor falso y no se ha definido fechaFin.

Cardinalidad: 0-1.

InterfaseTiempo.fechaFin

tipo	xs:date
código fuente	<xs:element name=" fechaFin " type="xs:date" minOccurs="0"/>

Descripción: Indica la fecha de cierre de un objeto; el significado de dicho cierre depende del LMS, por ejemplo, puede ser el último día en que los alumnos tienen acceso, o el día de resolución de una actividad, etc. Este dato es requerido cuando esFechaAbierta tiene valor falso y no se ha definido fechaInicio.

Cardinalidad: 0-1.

InterfaseTiempo.duracion

tipo	xs:float
código fuente	<xs:element name="duracion" type="xs:float" minOccurs="0"/>

Descripción: Indica el período de vigencia de un objeto; el significado de dicha vigencia depende del LMS. Este dato es opcional y útil para afinar la vigencia cuando esFechaAbierta tiene valor verdadero. Un caso es poner a disposición del usuario un recurso por cierto período a partir de que lo accesa por primera vez, sin importar en qué día en particular ocurre este evento, ni si coincide con otros usuarios.

Cardinalidad: 0-1.

InterfaseTiempo.unidadDuracion

tipo	restricción de xs:string
facets	enumeration Mes enumeration Dia enumeration Hora
código fuente	<pre><xs:element name="unidadDuracion" minOccurs="0"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Mes"/> <xs:enumeration value="Dia"/> <xs:enumeration value="Hora"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

Descripción: unidadDuracion es una enumeración para interpretar el valor que se asignó a duracion; los valores que se ofrecen abarcan las opciones usadas comúnmente y puede no ser necesario extenderlos. Este elemento es requerido si duracion es diferente de 0 (en realidad, mayor de 0).

Cardinalidad: 0-1.

Documentación del esquema ContextosEvaluacion (Evaluador)

Archivo: ContextosEvaluacion.xsd

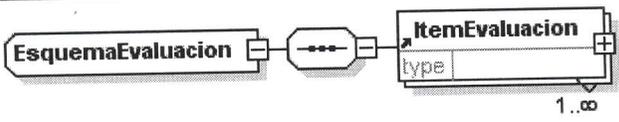
Este esquema no tiene un elemento root propiamente, ya que sirve de repositorio de los elementos e interfases que implementan el contexto de evaluación, a saber:

Elementos:

- EsquemaEvaluacion
- ItemEvaluacion
- InterfaseAplicacion
- InterfaseComputo

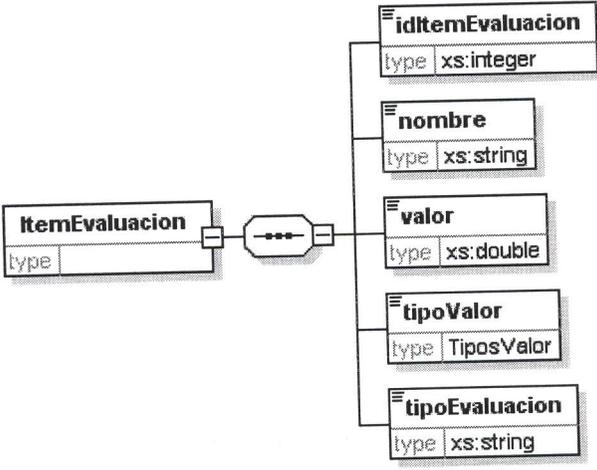
- TiposAccesoARespuesta
- TiposValor
- FormulasComputo

EsquemaEvaluacion

diagrama	
código fuente	<pre><xs:complexType name="EsquemaEvaluacion"> <xs:sequence> <xs:element ref="ItemEvaluacion" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType></pre>

Descripción: EsquemaEvaluacion es en realidad una agrupación virtual de objetos ItemEvaluacion, que son los que concretamente se asocian con el contexto de evaluación.

EsquemaEvaluacion.ItemEvaluacion

diagrama	
código fuente	<pre><xs:element name="ItemEvaluacion"> <xs:complexType> <xs:sequence> <xs:element name="idItemEvaluacion" type="xs:integer"/> <xs:element name="nombre" type="xs:string"/></pre>

```

<xs:element name="valor" type="xs:double"/>
<xs:element name="tipoValor" type="TiposValor"/>
<xs:element name="tipoEvaluacion">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Objetiva"/>
      <xs:enumeration value="Subjetiva"/>
      <xs:enumeration value="Votacion"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

Descripción: ItemEvaluacion es el objeto que define de manera abstracta un tipo de actividades de evaluación o un segmento del curso. El conjunto de estos objetos conforman lo que se conoce como esquema de evaluación del curso.

Cardinalidad: 1

EsquemaEvaluacion.ItemEvaluacion.idItemEvaluacion

tipo	xs:integer
código fuente	<xs:element name="idItemEvaluacion" type="xs:integer"/>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1

EsquemaEvaluacion.ItemEvaluacion.nombre

tipo	xs:string
código fuente	<xs:element name="nombre" type="xs:string"/>

Descripción: El nombre del ítem de evaluación es arbitrario y sólo informativo, aunque debe reflejar la intención de su uso. Debido a esto, pueden crearse tantos ítemes de evaluación en un curso como se requieran, sin importar si tal nombre existe en otro curso y tiene otro significado.

Cardinalidad: 1

EsquemaEvaluacion.ItemEvaluacion.valor

tipo	xs:double
código fuente	<xs:element name="valor" type="xs:double"/>

Descripción: valor es la proporción de la calificación final del curso que se asigna a todas las actividades de evaluación que implementen este ítem de evaluación en particular.

Cardinalidad: 1

EsquemaEvaluacion.ItemEvaluacion.tipoValor

tipo	TiposValor
facets	enumeration Porcentaje enumeration Puntuacion
código fuente	<xs:element name="tipoValor" type="TiposValor"/>

Descripción: tipoValor especifica cómo debe interpretarse el elemento valor del ítem de evaluación. Implementa la enumeración TiposValor (ver más adelante).

Cardinalidad: 1

EsquemaEvaluacion.ItemEvaluacion.tipoEvaluacion

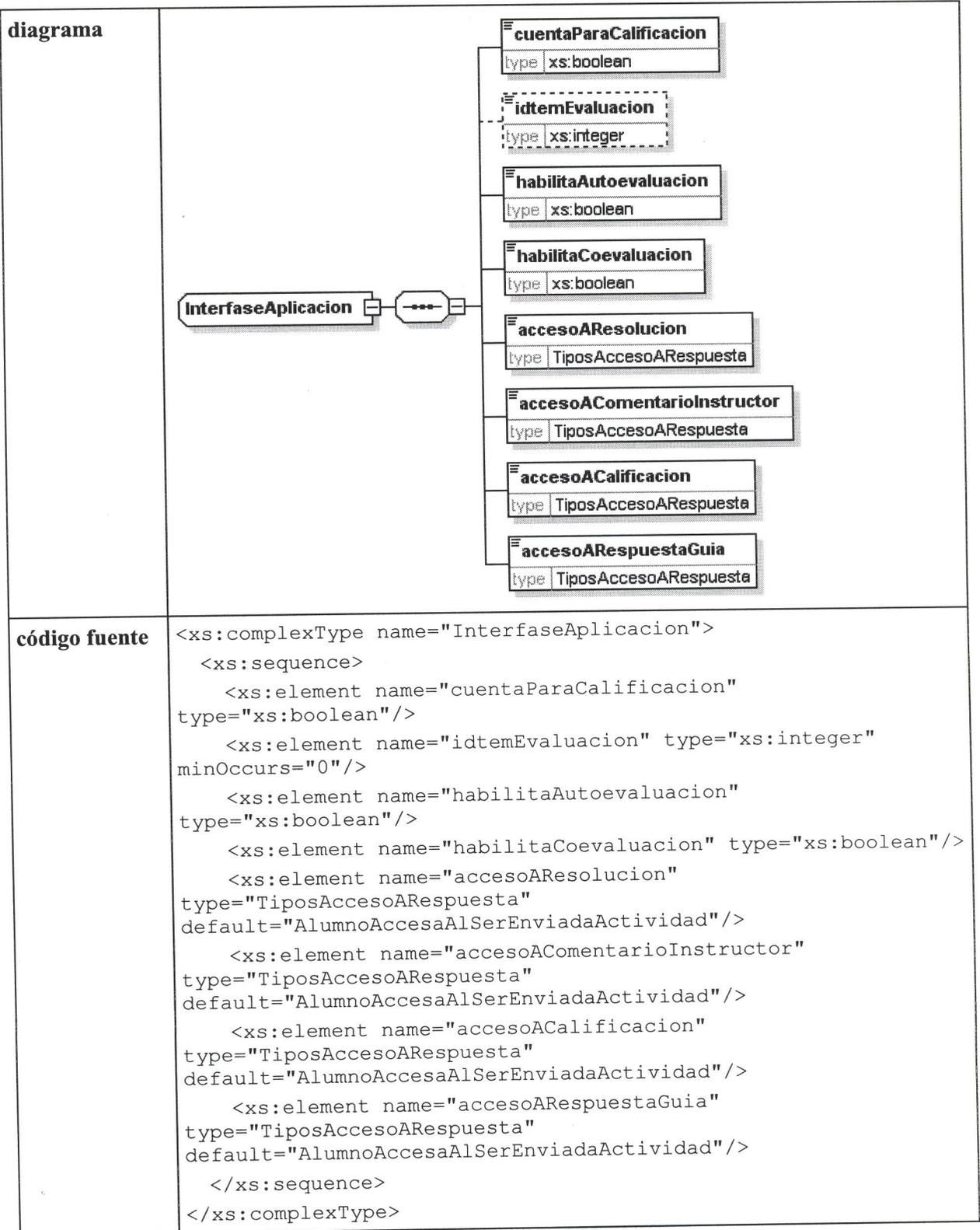
tipo	restricción de xs:string
facets	enumeration Objetiva enumeration Subjetiva enumeration Votacion

código fuente	<pre><xs:element name="tipoEvaluacion"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Objetiva"/> <xs:enumeration value="Subjetiva"/> <xs:enumeration value="Votacion"/> </xs:restriction> </xs:simpleType> </xs:element></pre>
----------------------	--

Descripción: tipoEvaluacion es una enumeración que define la implementación general que tiene este ítem de evaluación. Las opciones que se ofrecen son Objetiva, para evaluar resoluciones explícitas de una actividad; Subjetiva, para asignar notas a los alumnos, sin que medie una resolución explícita de su parte (por ejemplo, cuando se desea evaluar una actitud); y Votación, para que una actividad se evaluada por un conjunto en forma de encuesta. Esta enumeración puede extenderse a otros tipos válidos para el sistema.

Cardinalidad: 1

InterfaseAplicacion



Descripción: Esta interfase, junto con la interfase de cómputo, conforma el contexto de evaluación. InterfaseAplicacion define la forma en que se implementará una actividad de evaluación, así como el acceso a las respuestas de un alumno a la misma.

InterfaseAplicacion.cuentaParaCalificacion

tipo	xs:boolean
código fuente	<xs:element name="cuentaParaCalificacion" type="xs:boolean"/>

Descripción: Este elemento booleano especifica si el resultado de una actividad repercutirá en la calificación del curso. Al utilizarlo con valor falso en realidad se indica que la actividad en cuestión es diagnóstica, o tiene un propósito puramente formativo.

Cardinalidad: 1

InterfaseAplicacion.idItemEvaluacion

tipo	xs:boolean
código fuente	<xs:element name="idtemEvaluacion" type="xs:integer" minOccurs="0"/>

Descripción: Referencia al ítem de evaluación asociado con una actividad de evaluación, puesto que el atributo anterior, cuentaParaCalificacion, puede ser falso, el ítem de evaluación no sería requerido, de ahí que se trate de un valor que puede ser 0 o nulo.

Cardinalidad: 0-1

InterfaseAplicacion.habilitaAutoevaluacion

tipo	xs:boolean
código fuente	<xs:element name="habilitaAutoevaluacion" type="xs:boolean"/>

Descripción: `habilitaAutoevaluacion` indica al sistema si el autor de una respuesta puede evaluarse a sí mismo. El grado o extensión en que esta autoevaluación puede ocurrir depende del LMS.

Cardinalidad: 1

InterfaseAplicacion.habilitaCoevaluacion

tipo	<code>xs:boolean</code>
código fuente	<code><xs:element name="habilitaCoevaluacion" type="xs:boolean"/></code>

Descripción: `habilitaCoevaluacion` indica al sistema si otros alumnos del curso, compañeros actuales del autor de una respuesta pueden evaluar dicha respuesta. El grado o extensión en que esta coevaluación puede ocurrir depende del LMS.

Cardinalidad: 1

InterfaseAplicacion.accesoAResolucion

tipo	<code>TiposAccesoARespuesta</code>
código fuente	<code><xs:element name="accesoAResolucion" type="TiposAccesoARespuesta"/></code>

Descripción: `accesoAResolucion` es una enumeración que determina quién y cuándo se puede acceder la respuesta a una actividad de evaluación. Implementa la enumeración `TiposAccesoARespuesta` (ver más adelante).

Cardinalidad: 1

InterfaseAplicacion.accesoAComentarioInstructor

tipo	<code>TiposAccesoARespuesta</code>
código fuente	<code><xs:element name="accesoAComentarioInstructor" type="TiposAccesoARespuesta"/></code>

Descripción: accesoAComentarioInstructor es una enumeración que determina quién y cuándo se puede acceder el comentario evaluador que ha hecho el instructor a la respuesta a una actividad de evaluación. Implementa la enumeración TiposAccesoARespuesta (ver más adelante).

Cardinalidad: 1

InterfaseAplicacion.accesoACalificacion

tipo	TiposAccesoARespuesta
código fuente	<pre><xs:element name="accesoACalificacion" type="TiposAccesoARespuesta"/></pre>

Descripción: accesoACalificacion es una enumeración que determina quién y cuándo se puede acceder a la calificación que ha recibido la respuesta a una actividad de evaluación. Implementa la enumeración TiposAccesoARespuesta (ver más adelante).

Cardinalidad: 1

InterfaseAplicacion.accesoARespuestaGuia

tipo	TiposAccesoARespuesta
código fuente	<pre><xs:element name="accesoARespuestaGuia" type="TiposAccesoARespuesta"/></pre>

Descripción: accesoARespuestaGuia es una enumeración que determina quién y cuándo se puede acceder a la respuesta guía, si existe, que el instructor ha elaborado para una actividad de evaluación. Implementa la enumeración TiposAccesoARespuesta (ver más adelante).

Cardinalidad: 1

InterfaseComputo.esOpcional

tipo	xs:boolean
código fuente	<code><xs:element name="esOpcional" type="xs:boolean"/></code>

Descripción: Atributo booleano que indica si la resolución del objeto implementador es opcional para el alumno o no.

Cardinalidad: 1

InterfaseComputo.FormulaComputo

tipo	restricción de xs:string
facets	enumeration Bruta enumeration Promedio enumeration DesviacionStandard enumeration z enumeration Percentiles
código fuente	<pre><xs:simpleType name="FormulasComputo"> <xs:restriction base="xs:string"> <xs:enumeration value="Bruta"/> <xs:enumeration value="Promedio"/> <xs:enumeration value="DesviacionStandard"/> <xs:enumeration value="z"/> <xs:enumeration value="Percentiles"/> </xs:restriction> </xs:simpleType></pre>

Descripción: FormulaComputo es una enumeración de las opciones de cómputo que ofrece el evaluador. Aquí se presentan algunas comunes, pero podría la lista podría adaptarse a las necesidades de la implementación en particular. Puesto que se asume que hay una fórmula por default para el sistema, se considera que este dato puede ser opcional.

Cardinalidad: 0-1

InterfaseComputo.valor

tipo	xs:double
código fuente	<xs:element name="valor" type="xs:double"/>

Descripción: Este elemento representa el valor bruto que tiene el objeto implementador.

Cardinalidad: 1

InterfaseComputo.tipoValor

tipo	TiposValor
código fuente	<xs:element name="tipoValor" type="TiposValor"/>

Descripción: tipoValor especifica cómo debe interpretarse el elemento valor del objeto implementador de InterfaseComputo. Implementa la enumeración TiposValor (ver más adelante).

Cardinalidad: 1

InterfaseComputo.distribucionValor

tipo	restricción de xs:string
facets	enumeration Auto enumeration Manual
código fuente	<pre><xs:element name="distribucionValor" minOccurs="0"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="Auto"/> <xs:enumeration value="Manual"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

Descripción: *distribucionValor* indica cómo debe distribirse el valor de una actividad entre sus subítemes, si los hay; bien sea que el sistema lo haga automáticamente (Auto) o que el instructor lo realice (Manual).

Cardinalidad: 0-1

TiposAccesoARespuesta

tipo	restricción de xs:string
facets	enumeration AlumnoAccesaAlSerEnviadaActividad enumeration AlumnoAccesaAlSerCalificadaActividad enumeration EquipoAccesaAlSerEnviadaActividad enumeration EquipoAccesaAlSerCalificadaActividad enumeration ClaseAccesaAlSerEnviadaActividad enumeration ClaseAccesaAlSerCalificadaActividad
código fuente	<pre> <xs:simpleType name="TiposAccesoARespuesta"> <xs:restriction base="xs:string"> <xs:enumeration value="AlumnoAccesaAlSerEnviadaActividad"/> <xs:enumeration value="AlumnoAccesaAlSerCalificadaActividad"/> <xs:enumeration value="EquipoAccesaAlSerEnviadaActividad"/> <xs:enumeration value="EquipoAccesaAlSerCalificadaActividad"/> <xs:enumeration value="ClaseAccesaAlSerEnviadaActividad"/> <xs:enumeration value="ClaseAccesaAlSerCalificadaActividad"/> </xs:restriction> </xs:simpleType> </pre>

Descripción: *TiposAccesoARespuesta* es un tipo global que ofrece una enumeración para controlar accesos a elementos relacionados con una actividad de evaluación y las respuestas a ella de los alumnos. Los valores son: *AlumnoAccesaAlSerEnviadaActividad*, permite al autor de la respuesta acceder su propia respuesta desde el momento en que se envía; *AlumnoAccesaAlSerCalificadaActividad*, permite al autor de la respuesta acceder su propia respuesta hasta que el instructor la califica; *EquipoAccesaAlSerEnviadaActividad*, permite al autor de la respuesta y a compañeros de su equipo, si pertenece a un subgrupo dentro del

curso, acceder la respuesta en cuestión desde el momento en que se envía;

EquipoAccesaAlSerCalificadaActividad, permite al autor de la respuesta y a compañeros de su equipo, si pertenece a un subgrupo dentro del curso, acceder la respuesta en cuestión hasta que es calificada; ClaseAccesaAlSerEnviadaActividad, permite al autor de la respuesta y a cualquier otro compañero actual del curso acceder la respuesta en cuestión desde el momento en que se envía; ClaseAccesaAlSerCalificadaActividad, permite al autor de la respuesta y a cualquier otro compañero actual del curso acceder la respuesta en cuestión hasta que es calificada. Esta enumeración puede extenderse con otro valores válidos para el LMS en particular.

TiposValor

tipo	restricción de xs:string
facets	enumeration Porcentaje enumeration Puntuacion
código fuente	<pre><xs:simpleType name="TiposValor"> <xs:restriction base="xs:string"> <xs:enumeration value="Porcentaje"/> <xs:enumeration value="Puntuacion"/> </xs:restriction> </xs:simpleType></pre>

Descripción: La enumeración TiposValor ofrece opciones para interpretar elementos valor en los objetos que implementen la interfase de cómputo. Su valores son Porcentaje, para interpretar valor como porcentaje de un valor superior (por ejemplo, el valor de un ítem de evaluación sería porcentaje de la calificación final; el valor del ítem de una actividad sería un porcentaje del valor total de la actividad, la cual a su vez es relativa al ítem de evaluación); y Puntuacion, para interpretar valor como puntos brutos. Este elemento puede extenderse con otros valores válidos para el sistema en particular.

Documentación del esquema Calificaciones (Evaluador)

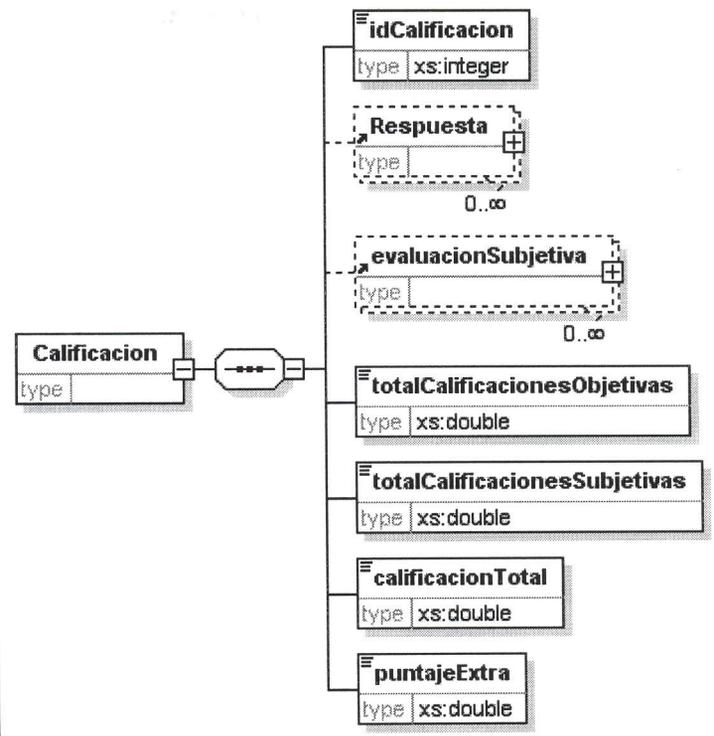
Archivo: Calificaciones.xsd

Este es el esquema que desarrolla prácticamente el evaluador y, por tanto, concentra los elementos propios de la propuesta de estándar que ofrece esta investigación.

Elementos:

- Calificacion
- Respuesta
- RespuestaItem
- Coevaluacion
- Votacion
- EvaluacionSubjetiva
- EvaluacionType

Calificacion

<p>diagrama</p>	
<p>código fuente</p>	<pre> <xs:element name="Calificacion"> <xs:complexType> <xs:sequence> <xs:element name="idCalificacion" type="xs:integer"/> <xs:element ref="Respuesta" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="evaluacionSubjetiva" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="totalCalificacionesObjetivas" type="xs:double" default="0"/> <xs:element name="totalCalificacionesSubjetivas" type="xs:double" default="0"/> <xs:element name="calificacionTotal" type="xs:double"/> <xs:element name="puntajeExtra" type="xs:double"/> </xs:sequence> </xs:complexType> </xs:element> </pre>

Descripción: Este es un objeto generado por el sistema evaluador; el instructor sólo puede cambiar aquí el puntaje extra, y éste dependerá de las políticas de implementación del curso que tenga el LMS.

Calificacion.idCalificacion

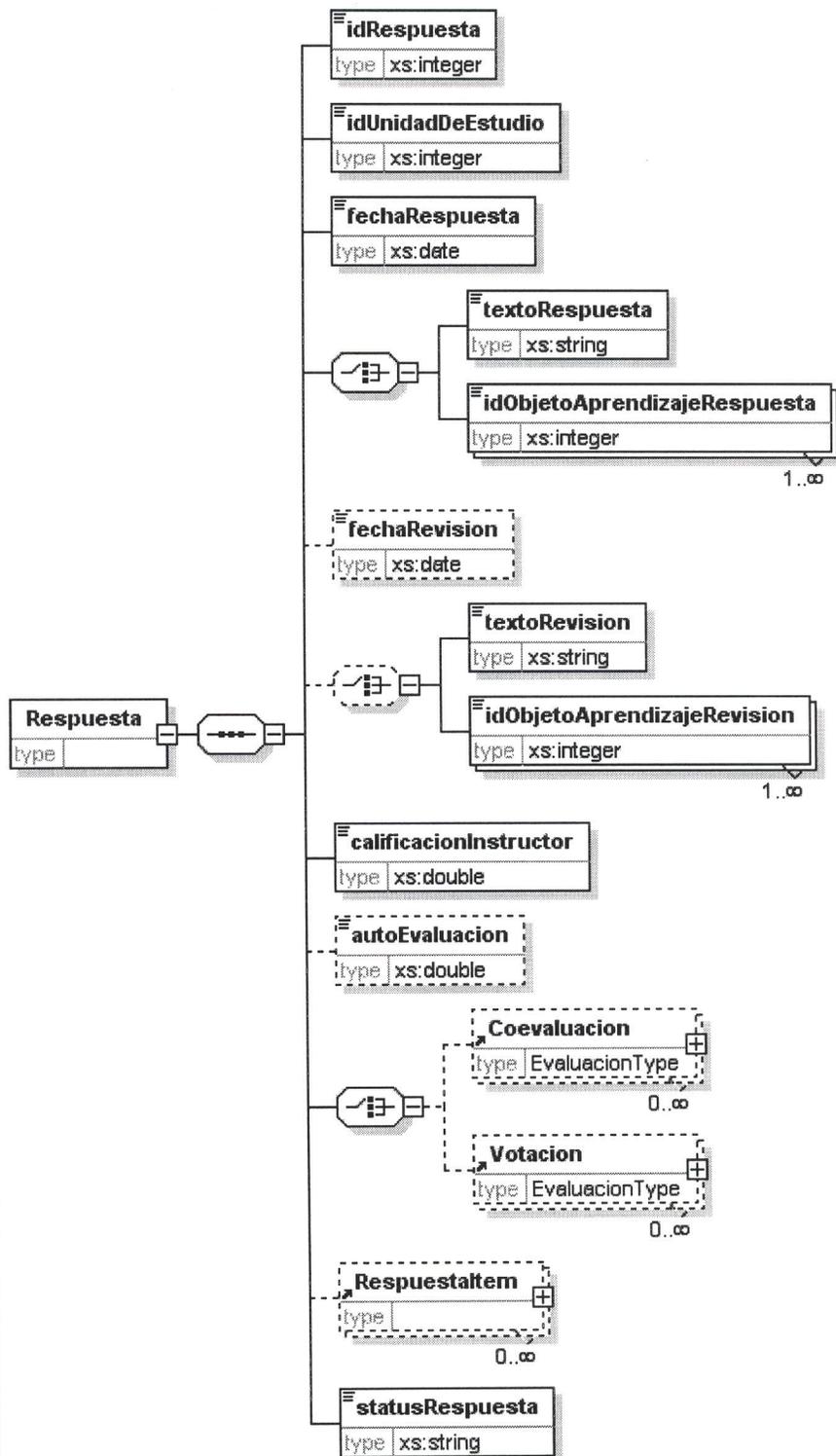
Tipo	xs:integer
código fuente	<xs:element name="idCalificacion" type="xs:integer"/>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1

Calificacion.Respuesta

diagrama



código fuente

```

<xs:element name="Respuesta">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="idRespuesta" type="xs:integer"/>
    
```

```

<xs:element name="idUnidadDeEstudio" type="xs:integer"/>
<xs:element name="fechaRespuesta" type="xs:date"/>
<xs:choice>
  <xs:element name="textoRespuesta" type="xs:string"/>
  <xs:element name="idObjetoAprendizajeRespuesta"
type="xs:integer" maxOccurs="unbounded"/>
</xs:choice>
<xs:element name="fechaRevision" type="xs:date"
minOccurs="0"/>
<xs:choice minOccurs="0">
  <xs:element name="textoRevision" type="xs:string"/>
  <xs:element name="idObjetoAprendizajeRevision"
type="xs:integer" maxOccurs="unbounded"/>
</xs:choice>
<xs:element name="calificacionInstructor"
type="xs:double" default="0"/>
<xs:element name="autoEvaluacion" type="xs:double"
minOccurs="0"/>
<xs:choice>
  <xs:element ref="Coevaluacion" minOccurs="0"
maxOccurs="unbounded"/>
  <xs:element ref="Votacion" minOccurs="0"
maxOccurs="unbounded"/>
</xs:choice>
<xs:element ref="RespuestaItem" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="statusRespuesta">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Enviada"/>
      <xs:enumeration value="Revisada"/>
      <xs:enumeration value="Rechazada"/>
      <xs:enumeration value="Aprobada"/>
      <xs:enumeration value="Reprogramada"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

Descripción: Cada objeto Respuesta representa la resolución a una actividad de evaluación, por parte de un alumno específico. A partir de ella, dependiendo del contexto de

evaluación de la actividad, pueden crearse autoevaluaciones y coevaluaciones, y sus datos correspondientes.

Cardinalidad: 0-∞.

Calificacion.Respuesta.idRespuesta

tipo	xs:integer
código fuente	<xs:element name="idRespuesta" type="xs:integer"/>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Calificacion.Respuesta.idUnidadDeEstudio

tipo	xs:integer
código fuente	<xs:element name="idUnidadDeEstudio" type="xs:integer"/>

Descripción: Este elemento de tipo entero (integer) hace referencia a la unidad de estudio (transformada en actividad de evaluación al aplicársele un contexto de evaluación) que fue resuelta.

Cardinalidad: 1.

Calificacion.Respuesta.fechaRespuesta

tipo	xs:date
código fuente	<xs:element name="fechaRespuesta" type="xs:date"/>

Descripción: Dato de sólo lectura que genera el sistema cuando se graba una respuesta.

Cardinalidad: 1.

Calificacion.Respuesta.textoRespuesta

tipo	xs:string
código fuente	<xs:element name="textoRespuesta" type="xs:string"/>

Descripción: Campo de texto para la respuesta que resuelve una actividad. Este campo es requerido cuando no existe objetoRespuesta.

Cardinalidad: 1.

Calificacion.Respuesta.idObjetoAprendizajeRespuesta

tipo	xs:integer
código fuente	<xs:element name="idObjetoAprendizajeRespuesta" type="xs:integer" maxOccurs="unbounded"/>

Descripción: Referencia a un objeto de aprendizaje (ver el esquema correspondiente antes) que el alumno podría usar para resolver una actividad de evaluación. Por ejemplo, en lugar de escribir en un campo de texto una respuesta, podría enviar un archivo de texto, una imagen, un url, etc. Los objetos disponibles dependen del LMS. Este campo es requerido cuando no existe textoRespuesta.

Cardinalidad: 1.

Calificacion.Respuesta.fechaRevision

tipo	xs:date
código fuente	<xs:element name="fechaRevision" type="xs:date" minOccurs="0"/>

Descripción: fechaRevision registra la fecha en que, para un LMS en particular, una actividad se considera revisada por el instructor; podría ser, por ejemplo, al calificarla, aunque

en otros casos puede ser útil que la fecha corresponda al momento en que se abra la respuesta, sea que se califique o no.

Cardinalidad: 0-1.

Calificacion.Respuesta.textoRevision

tipo	xs:string
código fuente	<xs:element name="textoRevision" type="xs:string"/>

Descripción: Texto para observaciones, correcciones o instrucciones adicionales del instructor dirigidas al autor de la respuesta. Este campo es requerido cuando no existe objetoRevision.

Cardinalidad: 0-1.

Calificacion.Respuesta.idObjetoAprendizajeRevision

tipo	xs:integer
código fuente	<xs:element name="idObjetoAprendizajeRevision" type="xs:integer" maxOccurs="unbounded"/>

Descripción: Objeto de aprendizaje que el instructor pone a disposición del autor de la respuesta, como parte de la revisión de la misma. Por ejemplo, en lugar de escribir en un campo de texto una respuesta, podría enviar un archivo de texto, una imagen, un url, etc. Los objetos disponibles dependen del LMS. Este campo es requerido cuando no existe textoRevision.

Cardinalidad: 0-1.

Calificacion.Respuesta.calificacionInstructor

tipo	xs:double
-------------	-----------

código fuente	<code><xs:element name="calificacionInstructor" type="xs:double" default="0"/></code>
----------------------	---

Descripción: Valor numérico que el instructor asigna a la respuesta. Si el LMS utiliza un esquema de notas no numéricas, se recomienda no sobrescribir este campo, sino crear una capa extra de conversión, para conservar la interoperabilidad.

Cardinalidad: 1.

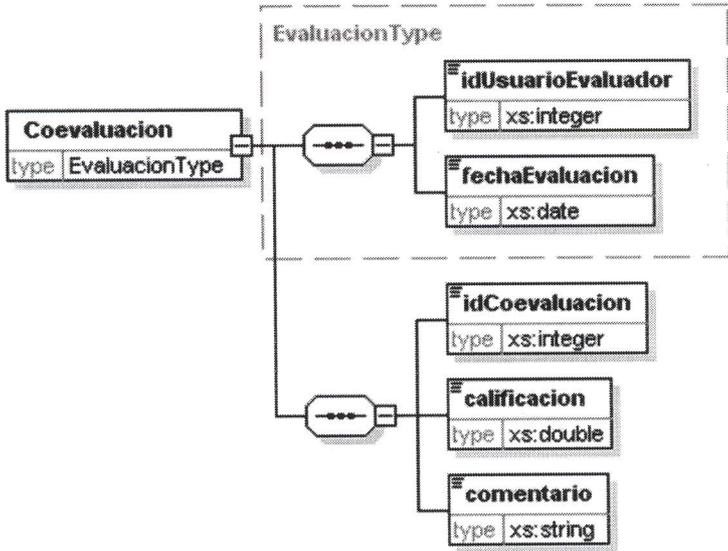
Calificacion.Respuesta.autocalificacion

tipo	xs:double
código fuente	<code><xs:element name="autoCalificacion" type="xs:double" minOccurs="0"/></code>

Descripción: Valor numérico que el alumno asigna a su propia respuesta, cuando la autoevaluación está habilitada. Si el LMS utiliza un esquema de notas no numéricas, se recomienda no sobrescribir este campo, sino crear una capa extra de conversión, para conservar la interoperabilidad.

Cardinalidad: 0-1.

Calificacion.Respuesta.coevaluacion

diagrama	
tipo	Extensión de EvaluacionType
código fuente	<pre> <xs:element name="Coevaluacion"> <xs:complexType> <xs:complexContent> <xs:extension base="EvaluacionType"> <xs:sequence> <xs:element name="idCoevaluacion" type="xs:integer"/> <xs:element name="calificacion" type="xs:double"/> <xs:element name="comentario" type="xs:string"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element> </pre>

Descripción: Cada coevaluacion es una evaluación que un alumno ha hecho de la respuesta de un compañero, cuando la coevaluación está habilitada. Es una extensión de la interfase EvaluacionType, que se explica más adelante. Para que exista este registro, el ítem de evaluación que sirvió de base para la actividad (unidad de estudio) que resuelve esta respuesta, debe ser de tipo objetivo y la coevaluación debe estar habilitada. Dicho tipo de ítem de evaluación implica que, para una respuesta dada, sólo existen potencialmente coevaluaciones abiertas, como esta, y no votaciones.

Cardinalidad: 0-∞

Calificacion.Respuesta.Coevaluacion.idCoevaluacion

tipo	xs:double
código fuente	<xs:element name="idCoevaluacion" type="xs:integer"/>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Calificacion.Respuesta.Coevaluacion.calificacion

tipo	xs:double
código fuente	<xs:element name="calificacion" type="xs:double"/>

Descripción: Nota asignada por el evaluador a la respuesta de un alumno.

Cardinalidad: 1.

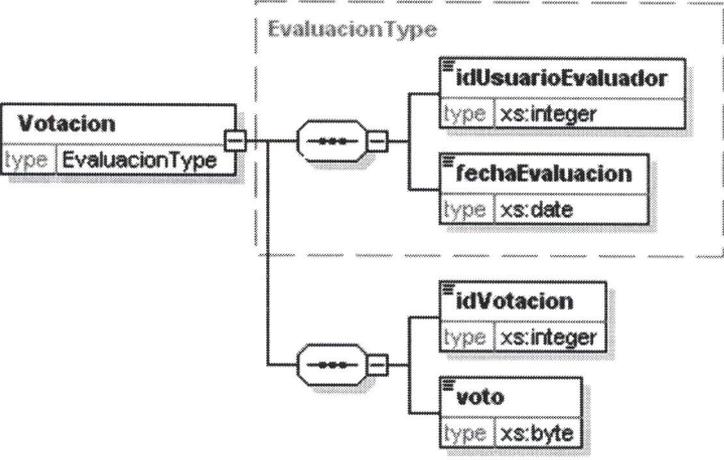
Calificacion.Respuesta.Coevaluacion.comentario

tipo	xs:string
código fuente	<xs:element name="comentario" type="xs:string"/>

Descripción: Texto para observaciones a la coevaluación.

Cardinalidad: 1.

Calificacion.Respuesta.Votacion

<p>diagrama</p>	
<p>tipo</p>	<p>Extensión de EvaluacionType</p>
<p>código fuente</p>	<pre> <xs:element name="Votacion"> <xs:complexType> <xs:complexContent> <xs:extension base="EvaluacionType"> <xs:sequence> <xs:element name="idVotacion" type="xs:integer"/> <xs:element name="voto" type="xs:byte"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element> </pre>

Descripción: Cada Votacion es una evaluación que un alumno ha hecho de la respuesta de un compañero, cuando la coevaluación está habilitada, eligiendo entre un conjunto definido de valores (por ejemplo, se trata de votar calificando la respuesta con un número entero entre 0 y 9). Es una extensión de la interfase EvaluacionType, que se explica más adelante. Para que exista este registro, el ítem de evaluación que sirvió de base para la actividad (unidad de estudio) que resuelve esta respuesta, debe ser de tipo votación y la coevaluación debe estar habilitada. Dicho tipo de ítem de evaluación implica que, para una respuesta dada, sólo existen

potencialmente votaciones, como ésta, y no coevaluaciones abiertas como las que se vieron anteriormente.

Cardinalidad: 0-∞

Calificacion.Respuesta.Votacion.idVotacion

tipo	EvaluacionType
código fuente	<code><xs:element name="idVotacion" type="xs:integer"/></code>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

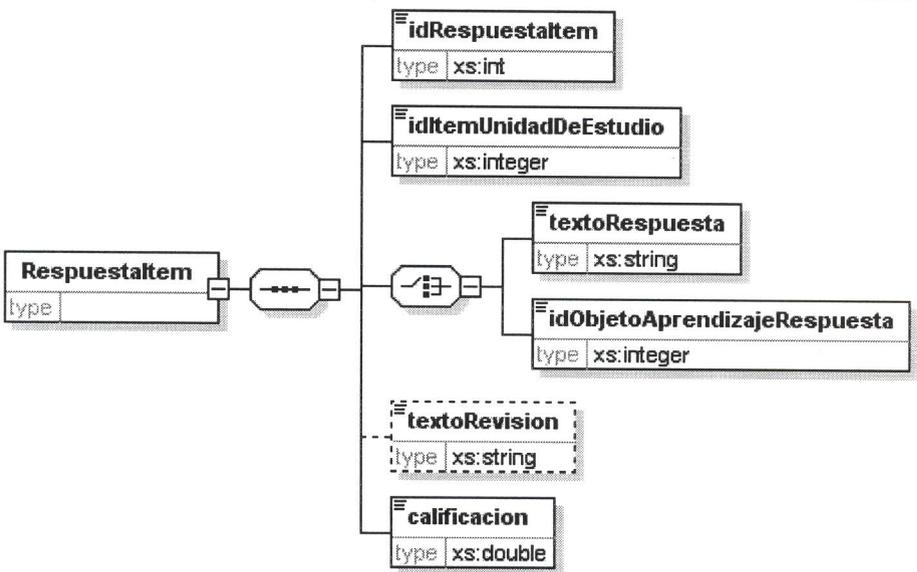
Calificacion.Respuesta.Votacion.voto

tipo	EvaluacionType
código fuente	<code><xs:element name="voto" type="xs:byte"/></code>

Descripción: Valor elegido entre una serie predefinida para calificar la respuesta a la que pertenece esta votación.

Cardinalidad: 1.

Calificacion.Respuesta.RespuestaItem

diagrama	
código fuente	<pre> <xs:element name="RespuestaItem"> <xs:complexType> <xs:sequence> <xs:element name="idRespuestaItem" type="xs:int"/> <xs:element name="idItemUnidadDeEstudio" type="xs:integer"/> <xs:choice> <xs:element name="textoRespuesta" type="xs:string"/> <xs:element name="idObjetoAprendizajeRespuesta" type="xs:integer"/> </xs:choice> <xs:element name="textoRevision" type="xs:string" minOccurs="0"/> <xs:element name="calificacion" type="xs:double"/> </xs:sequence> </xs:complexType> </xs:element> </pre>

Descripción: RespuestaItem representa la respuesta a un ítem particular de la unidad evaluable, cuando tales ítems existen, por ejemplo el objeto Respuesta puede representar la resolución global de un examen y RespuestaItem la resolución de una pregunta de dicho examen.

Cardinalidad: 0-∞.

Calificacion.Respuesta.RespuestaItem.idRespuestaItem

tipo	xs:integer
código fuente	<xs:element name="idRespuestaItem" type="xs:int"/>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Calificacion.Respuesta.RespuestaItem.idItemUnidadDeEstudio

tipo	xs:integer
código fuente	<xs:element name="idItemUnidadDeEstudio" type="xs:integer"/>

Descripción: Referencia al identificador del ítem de la unidad que se está resolviendo.

Cardinalidad: 1.

Calificacion.Respuesta.RespuestaItem.textoRespuesta

tipo	xs:string
código fuente	<xs:element name="textoRespuesta" type="xs:string"/>

Descripción: Campo de texto de la respuesta del alumno a un ítem de una actividad de evaluación. Este campo es requerido cuando no existe un objetoRespuesta.

Cardinalidad: 0-1.

Calificacion.Respuesta.RespuestaItem.idObjetoAprendizajeRespuesta

tipo	xs:string
código fuente	<xs:element name="idObjetoAprendizajeRespuesta" type="xs:integer"/>

Descripción: Referencia a un objeto de aprendizaje (ver el esquema correspondiente antes) que el alumno podría usar para resolver una actividad de evaluación. Este campo es requerido cuando no existe textoRespuesta.

Cardinalidad: 0-1.

Calificacion.Respuesta.RespuestaItem.textoRevision

tipo	xs:string
código fuente	<xs:element name="textoRevision" type="xs:string" minOccurs="0"/>

Descripción: Comentario opcional del instructor a una RespuestaItem.

Cardinalidad: 0-1.

Calificacion.Respuesta.RespuestaItem.calificacion

tipo	xs:double
código fuente	<xs:element name="calificacion" type="xs:double"/>

Descripción: Nota numérica que RespuestaItem ha ganado. Para algunos casos puede ser un procedimiento automático asignarla, por ejemplo cuando el sistema califica preguntas de tipo Falso/Verdadero.

Cardinalidad: 1.

Calificacion.Respuesta.statusRespuesta

tipo	restricción de xs:string
facets	enumeration Enviada enumeration Revisada enumeration Rechazada enumeration Aprobada enumeration Reprogramada
código fuente	<xs:element name="statusRespuesta">

```

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Enviada"/>
    <xs:enumeration value="Revisada"/>
    <xs:enumeration value="Rechazada"/>
    <xs:enumeration value="Aprobada"/>
    <xs:enumeration value="Reprogramada"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>

```

Descripción: statusRespuesta es una enumeración con los estados que puede tener una respuesta en cuanto a su evaluación. La generación de este campo de sólo lectura depende de los procedimientos del LMS en particular. Las opciones son: Enviada, que es el estado inicial cuando se resuelve una actividad creándose una respuesta; Revisada, si la respuesta ha sido evaluada, total o parcialmente; Rechazada, aunque puede estar implícito este estado con una calificación reprobatoria, registrarlo explícitamente puede ser útil para verificar el cumplimiento de una actividad; Aprobada, es otro estado que podría estar implícito cuando la respuesta obtiene una calificación aprobatoria, pero que es útil explícitamente cuando, por ejemplo, las actividades tienen diferentes rangos de aprobación y se requiere saber en general cuáles han sido aprobadas, sin necesidad de con cuánto se ha logrado; Reprogramada, que es el estado que permite al alumno repetir una actividad, según el procedimiento establecido por el propio LMS. Esta enumeración puede extenderse a otros tipos válidos para el sistema en particular.

Cardinalidad: 1.

Calificacion.evaluacionSubjetiva

diagrama	
código fuente	<pre> <xs:element name="evaluacionSubjetiva"> <xs:complexType> <xs:sequence> <xs:element name="idEvaluacionSubjetiva" type="xs:integer"/> <xs:element name="idItemEvaluacion" type="xs:integer"/> <xs:element name="fechaEvaluacion" type="xs:date"/> <xs:element name="calificacion" type="xs:double"/> <xs:element name="comentario" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element> </pre>

Descripción: Cada elemento `evaluacionSubjetiva` corresponde a la calificación de un ítem subjetivo, que son aquellos que se producen por iniciativa del instructor, sin que medie resolución de actividad explícita por parte del alumno. Estas notas pueden ser útiles cuando quiere evaluarse la actitud o ciertos avances individuales.

Cardinalidad: 0-∞.

Calificacion.evaluacionSubjetiva.idEvaluacionSubjetiva

tipo	xs:integer
código fuente	<xs:element name="idEvaluacionSubjetiva" type="xs:integer"/>

Descripción: El identificador es de tipo entero (integer) y es un valor único del objeto en el sistema que se tome para identificarlo.

Cardinalidad: 1.

Calificacion.evaluacionSubjetiva.idItemEvaluacion

tipo	xs:integer
código fuente	<xs:element name="iidIemEvaluacion" type="xs:integer"/>

Descripción: Referencia al ítem de evaluación del que se deriva la calificación subjetiva.

Cardinalidad: 1.

Calificacion.evaluacionSubjetiva.fechaEvaluacion

tipo	xs:date
código fuente	<xs:element name="fechaEvaluacion" type="xs:date"/>

Descripción: fechaEvaluacion registra cuándo se ha generado el objeto Calificacion. El disparador de ese evento depende del LMS, así como verificar el rango de fechas válidos, que se esperaría estuviera restringido al contexto del curso en general.

Cardinalidad: 1.

Calificacion.evaluacionSubjetiva.calificacion

tipo	xs:double
código fuente	<xs:element name="calificacion" type="xs:double"/>

Descripción: Nota numérica que representa la evaluación subjetiva de este objeto Calificacion. El rango depende del LMS en particular. Para el caso en que se utilicen notas no

numéricas, se recomienda que el LMS agregue una capa de conversión, manteniendo calificación como numérica, para conservar la interoperabilidad.

Cardinalidad: 1.

Calificacion.evaluacionSubjetiva.comentario

tipo	xs:string
código fuente	<xs:element name="comentario" type="xs:string"/>

Descripción: Campo de texto que puede ser opcional o no, en el que el instructor puede agregar observaciones a la calificación obtenida.

Cardinalidad: 1.

Calificacion.totalCalificacionesObjetivas

Tipo	xs:double
código fuente	<xs:element name="totalCalificacionesObjetivas" type="xs:double" default="0"/>

Descripción: Este es el registro de la calificación resultante de las actividades que fueron evaluadas objetivamente, tanto por calificación abierta de una respuesta (calificación del instructor, autoevaluación, si existe, y coevaluaciones, si existen) como por votación, a las que se les ha aplicado la fórmula correspondiente de cómputo, si fue elegida una explícita (de lo contrario, el LMS deberá implementar alguna por default, que puede ser simplemente la suma de resultados brutos). Se trata de un campo de sólo lectura.

Cardinalidad: 1

Calificacion.totalCalificacionSubjetivas

tipo	xs:double
código fuente	<xs:element name="totalCalifiicacionSubjetivas" type="xs:double" default="0"/>

Descripción: Registro de la calificación resultante de las actividades que fueron evaluadas subjetivamente, tras aplicar la fórmula correspondiente de cómputo, si fue elegida una explícita (de lo contrario, el LMS deberá implementar alguna por default, que puede ser simplemente la suma de resultados brutos). Se trata de un campo de sólo lectura.

Cardinalidad: 1

Calificacion.calificacionTotal

tipo	xs:double
código fuente	<xs:element name="calificacionTotal" type="xs:double"/>

Descripción: Para obtener este elemento, se ponderan el conjunto de evaluaciones de un alumno (objetivas, votaciones y subjetivas) de acuerdo con la fórmula de cómputo elegida para el curso en general. Este campo es de sólo lectura.

Cardinalidad: 1

Calificacion.puntajeExtra

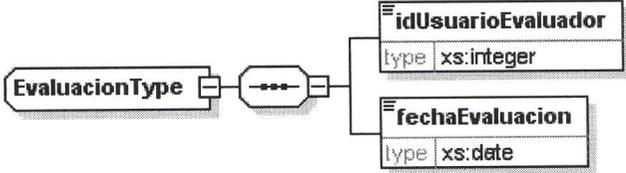
tipo	xs:double
código fuente	<xs:element name="puntajeExtra" type="xs:double"/>

Descripción: puntajeExtra es el único campo de lectura y escritura de la clase Calificacion. Registra los puntos adicionales que el instructor añadió a la calificación de un

alumno. Las restricciones en cuanto al uso de este campo (por ejemplo, su rango), deben ser especificadas por el LMS.

Cardinalidad: 1

EvaluacionType

diagrama	
código fuente	<pre><xs:complexType name="EvaluacionType"> <xs:sequence> <xs:element name="idUsuarioEvaluador" type="xs:integer"/> <xs:element name="fechaEvaluacion" type="xs:date"/> </xs:sequence> </xs:complexType></pre>

Descripción: Interfase para ser utilizado por los procedimientos que involucran varias evaluaciones para una misma respuesta, como en el caso de la coevaluación, abierta (objetiva) o por votación.

EvaluacionType.idUsuarioEvaluador

tipo	xs:integer
código fuente	<xs:element name="idUsuarioEvaluador" type="xs:integer"/>

Descripción: Este elemento de tipo entero (integer) representa una referencia al identificador (ID) del usuario que realiza la evaluación.

Cardinalidad: 1

EvaluacionType.fechaEvaluacion

tipo	xs:date
código fuente	<code><xs:element name="fechaEvaluacion" type="xs:date"/></code>

Descripción: Campo generado por el sistema que registra cuándo se ha realizado la evaluación.

Cardinalidad: 1

APÉNDICE C

EJEMPLOS DE APLICACIÓN DEL ESTÁNDAR

Código XML de la evaluación en un curso

El siguiente código corresponde a un archivo XML bien formado, producido por un sistema de evaluación del aprendizaje hipotético. Describe un curso llamado “Programación avanzada” con dos unidades de estudio, “Unidad I. Introducción” y “Unidad II. Datos primitivos de Visual Basic .Net”; ésta última es una actividad de evaluación.

En el curso están inscritos dos alumnos (sus identificadores de referencia o idUsuario son 1345 y 1560; datos adicionales, como su nombre, serían provistos por el LMS). Se incluyen datos de la resolución de la actividad que cada uno hizo.

El código fue positivamente validado por los esquemas de la propuesta de estándar; por tanto, puede afirmarse que el sistema que lo ha producido cumple con el estándar.

```
<?xml version="1.0" encoding="UTF-8"?>
<Curso xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Cursos.xsd">
  <idCurso>23</idCurso>
  <datosCurso>
    <relacion>
      <ubicacion>
        <idEstructuraSuperior>0</idEstructuraSuperior>
        <orden>1</orden>
        <nombre>Programación Avanzada</nombre>
      </ubicacion>
    </relacion>
    <tiempo>
      <esFechaAbierta>0</esFechaAbierta>
      <fechaInicio>2004-08-26</fechaInicio>
      <fechaFin>2004-12-13</fechaFin>
    </tiempo>
    <puntajeMaximo>100</puntajeMaximo>
    <puntajeMinimo>50</puntajeMinimo>
    <puntajeMinimoParaAprobar>70</puntajeMinimoParaAprobar>
  </datosCurso>
```

```

<instructor>
  <idUsuario>200</idUsuario>
</instructor>
<alumno>
  <idUsuario>1345</idUsuario>
  <fechaIngreso>2004-08-28</fechaIngreso>
  <status>Inscrito</status>
  <Calificacion>
    <idCalificacion>400</idCalificacion>
    <Respuesta>
      <idRespuesta>23450</idRespuesta>
      <idUnidadDeEstudio>1008</idUnidadDeEstudio>
      <fechaRespuesta>2004-09-10</fechaRespuesta>
      <textoRespuesta>Mi respuesta al ejercicio es bla bla bla
bla...</textoRespuesta>
      <fechaRevision>2004-09-12</fechaRevision>
      <textoRevision>Excelente tarea</textoRevision>
      <calificacionInstructor>100</calificacionInstructor>
      <autoEvaluacion>100</autoEvaluacion>
      <Coevaluacion>
        <idUsuarioEvaluador>1560</idUsuarioEvaluador>
        <fechaEvaluacion>2004-09-11</fechaEvaluacion>
        <idCoevaluacion>300</idCoevaluacion>
        <calificacion>90</calificacion>
        <comentario>Estuvo bien, pero no me gustó el resultado</comentario>
      </Coevaluacion>
      <statusRespuesta>Aprobada</statusRespuesta>
    </Respuesta>
    <totalCalificacionesObjetivas>95</totalCalificacionesObjetivas>
    <totalCalificacionesSubjetivas>0</totalCalificacionesSubjetivas>
    <calificacionTotal>95</calificacionTotal>
    <puntajeExtra>5</puntajeExtra>
  </Calificacion>
</alumno>
<alumno>
  <idUsuario>1560</idUsuario>
  <fechaIngreso>2004-08-28</fechaIngreso>
  <status>Inscrito</status>
  <Calificacion>
    <idCalificacion>0</idCalificacion>
    <Respuesta>
      <idRespuesta>45601</idRespuesta>
      <idUnidadDeEstudio>1008</idUnidadDeEstudio>
      <fechaRespuesta>2004-09-15</fechaRespuesta>
      <idObjetoAprendizajeRespuesta>456</idObjetoAprendizajeRespuesta>
      <fechaRevision>2004-09-15</fechaRevision>
      <textoRevision>Aunque está bien tu tarea, la enviaste tarde</textoRevision>
      <calificacionInstructor>85</calificacionInstructor>
      <autoEvaluacion>100</autoEvaluacion>
      <statusRespuesta>Aprobada</statusRespuesta>
    </Respuesta>
    <totalCalificacionesObjetivas>90</totalCalificacionesObjetivas>
    <totalCalificacionesSubjetivas>0</totalCalificacionesSubjetivas>
    <calificacionTotal>90</calificacionTotal>
    <puntajeExtra>0</puntajeExtra>
  </Calificacion>
</alumno>
<Objetivo>
  <idObjetivo>23</idObjetivo>
  <descripcion>Aprender Visual Basic .Net</descripcion>
</Objetivo>
<Seccion>

```

```

<idSeccion>12</idSeccion>
<relacion>
  <ubicacion>
    <idEstructuraSuperior>0</idEstructuraSuperior>
    <orden>1</orden>
    <nombre>Capítulo I</nombre>
  </ubicacion>
</relacion>
<tiempo>
  <esFechaAbierta>1</esFechaAbierta>
</tiempo>
<UnidadDeEstudio>
  <idUnidadDeEstudio>1003</idUnidadDeEstudio>
  <idObjetivo>23</idObjetivo>
  <idObjetoAprendizaje>760</idObjetoAprendizaje>
  <relacion>
    <ubicacion>
      <idEstructuraSuperior>0</idEstructuraSuperior>
      <orden>1</orden>
      <nombre>Unidad I. Introducción</nombre>
    </ubicacion>
  </relacion>
  <tiempo>
    <esFechaAbierta>1</esFechaAbierta>
  </tiempo>
</UnidadDeEstudio>
<UnidadDeEstudio>
  <idUnidadDeEstudio>1008</idUnidadDeEstudio>
  <idObjetivo>23</idObjetivo>
  <idObjetoAprendizaje>4567</idObjetoAprendizaje>
  <relacion>
    <ubicacion>
      <idEstructuraSuperior>0</idEstructuraSuperior>
      <orden>2</orden>
      <nombre>Unidad II. Datos primitivos de Visual Basic .Net</nombre>
    </ubicacion>
  </relacion>
  <tiempo>
    <esFechaAbierta>0</esFechaAbierta>
    <fechaInicio>2004-09-10</fechaInicio>
    <fechaFin>2004-09-10</fechaFin>
  </tiempo>
  <aplicacionEvaluacion>
    <cuentaParaCalificacion>1</cuentaParaCalificacion>
    <idtemEvaluacion>100</idtemEvaluacion>
    <habilitaAutoevaluacion>1</habilitaAutoevaluacion>
    <habilitaCoevaluacion>1</habilitaCoevaluacion>
    <accesoAResolucion>ClaseAccesaAlSerEnviadaActividad</accesoAResolucion>
  <accesoAComentarioInstructor>AlumnoAccesaAlSerEnviadaActividad</accesoAComentarioInstructor>
  <accesoACalificacion>AlumnoAccesaAlSerEnviadaActividad</accesoACalificacion>

  <accesoARespuestaGuia>ClaseAccesaAlSerCalificadaActividad</accesoARespuestaGuia>
  </aplicacionEvaluacion>
  <computo>
    <esOpcional>0</esOpcional>
    <valor>100</valor>
    <tipoValor>Puntuacion</tipoValor>
  </computo>
</UnidadDeEstudio>
</Seccion>
<EsquemaEvaluacion>

```

```

<ItemEvaluacion>
  <idItemEvaluacion>100</idItemEvaluacion>
  <nombre>Ejercicios</nombre>
  <valor>50</valor>
  <tipoValor>Porcentaje</tipoValor>
  <tipoEvaluacion>Objetiva</tipoEvaluacion>
</ItemEvaluacion>
<ItemEvaluacion>
  <idItemEvaluacion>340</idItemEvaluacion>
  <nombre>Lecturas</nombre>
  <valor>50</valor>
  <tipoValor>Porcentaje</tipoValor>
  <tipoEvaluacion>Objetiva</tipoEvaluacion>
</ItemEvaluacion>
</EsquemaEvaluacion>
</Curso>

```

Intercambio de datos entre dos sistemas

Ahora consideremos el caso de dos aplicaciones existentes de la Universidad de Montemorelos: El portal del maestro (pm), que ofrece una administración general de los grupos de alumnos y cuenta con la posibilidad de crear un esquema de evaluación, aunque no crea clases propiamente; y tenemos el e42, un LMS que también crea esquemas de evaluación, aunque este también crea los cursos, con contenidos y actividades de evaluación. Veamos cómo reportan las actividades de evaluación cada uno en las siguientes tablas (los datos se han adaptado, pero siguen muy de cerca la realidad):

Ejemplo de datos de evaluación del Portal del maestro

idMaestro	ClaveMat	Curso Carga ID	alumno	ítemEvaluación	calificación
9800000	CCSC1970CIII421	02031F-0000	871074	Tareas	90
9800000	CCSC1970CIII421	02031F-0000	871074	Lecturas	65

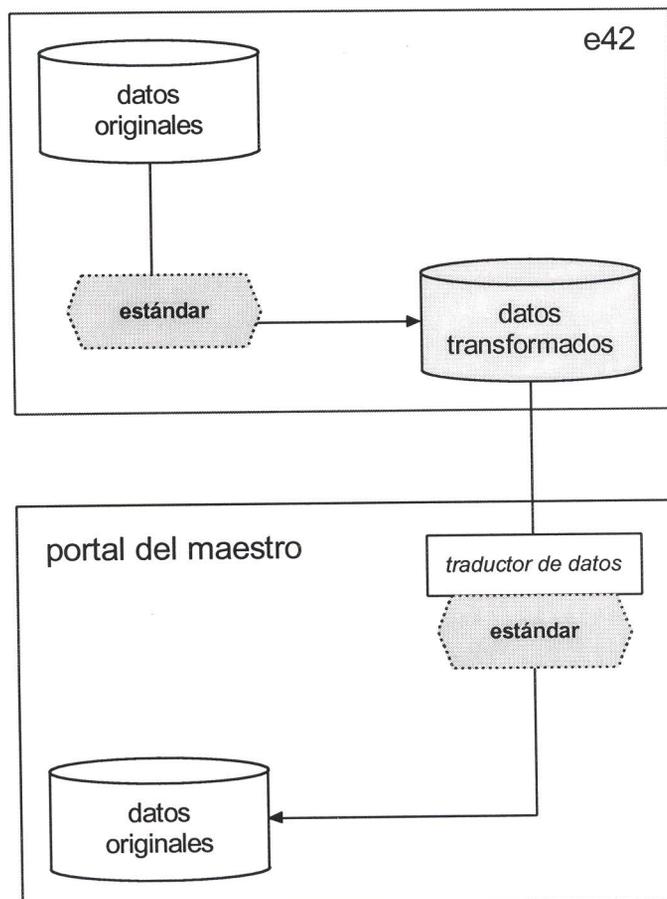
Ejemplo de datos de evaluación del e42

idMateria	idActividadEnviada	idActividad	alumno	texto	observaciones	calificación
350	8769	2345	user	bla bla bla	¡Excelente!	100
350	8870	2216	user	y otro bla bla	Debes mejorar	70
350	8941	2217	user	y más bla bla	Muy bien	100
350	9023	1630	user	de nuevo bla	Casi perfecto	90
350	9255	500	user	super bla bla	Hazlo de nuevo	50

El problema que se enfrenta es aprovechar la información del e42 para transmitirla al pm, donde se conservan los datos oficiales y finales del desempeño de un alumno. Además, se quiere lograr esto sin modificar ninguno de los dos sistemas y sin invertir mucho en la solución.

La solución puede ser precisamente la presente propuesta de estándar. Siguiéndolo el e42 puede adaptar sus datos ya existentes a las estructuras de datos definidas en el estándar; con eso bastaría para estar listos para ser transmitidos al pm; entonces éste, puede crear métodos que interpreten los datos, a partir del patrón que propone el estándar, para condensarlos en sus propias tablas.

Transformación de datos en el Portal del maestro y el e42



El resultado de transformar los datos del e42 al estándar sería una tabla de respuestas y una tabla de calificaciones:

Ejemplo de datos de respuestas del e42 transformados

identificador	idCalificacion	UnidadDeEstudio	textoRespuesta	textoRevision	calificacion	status
8769	45021	2345	bla bla bla	¡Excelente!	100	Revisada
8870	45021	2216	y otro bla bla	Debes mejorar	70	Revisada
8941	45021	2217	y más bla bla	Muy bien	100	Revisada
9023	45021	1630	de nuevo bla	Casi perfecto	90	Revisada
9255	45021	500	super bla bla	Hazlo de nuevo	50	Reprogramada

Ejemplo de calificaciones del e42 transformados

identificador	curso	alumno	calificacionesObjetivas	calificacionTotal	puntosExtras
45021	203	1205	77.5		0

Si ambos sistemas no mantienen un mismo catálogo de materias y alumnos, como se ve en el caso anterior, podrían requerirse algunas tablas auxiliares de equivalencias.

APÉNDICE D

RECURSOS WEB Y CD ANEXO

Estándares y documentos de organizaciones estandarizadoras o desarrolladoras internacionales

AICC: <http://www.aicc.org/pages/related.html>
ARIADNE: <http://ariadne.unil.ch/>
Carnegie Mellon Software Engineering Institute (SEI): <http://www.sei.cmu.edu>
Cetis: <http://www.cetis.ac.uk>
IEEE: <http://ltsc.ieee.org/>
IMS Global Learning Consortium: <http://www.imsproject.org/>
ISO: <http://www.iso.ch/iso/en/ISOOnline.openpage>
PROMETEUS: <http://www.prometeus.org/index.cfm?PID=315&language=5>
SIF: <http://www.sifinfo.org>
World Wide Web Consortium, <http://www.w3.org/Consortium>

Documentos sobre elearning y publicaciones electrónicas periódicas

Educational Technology Expertise Center, The Open University of Netherlands:
<http://learningnetworks.org>
Encuentros internacionales de Educación a Distancia de la Universidad de Guadalajara:
<http://innova.udg.mx/encuentro/index.cfm?seccion=I>
Estándares del Adventist Virtual Learning Network: <http://www.avln.org/standards.htm>
Learning objects & learning standards: <http://www.learnativity.com/standards.html>
Guía para la educación a distancia de la Universidad de Idaho, basada en el libro de Barry
Willis, *Distance Education - Strategies and Tools and Distance Education - A Practical Guide*:
<http://www.uidaho.edu/eo/distglan.html>
Guía para la enseñanza en línea de la Universidad de Minnesota:
<http://ltu.cce.umn.edu/guide/index.htm>
Quaderns Digitals .Net: <http://www.quadernsdigitals.net>
Red Iris: <http://edutec.rediris.es>
Revista Iberoamericana de Educación: <http://www.campus-oei.org/revista>
Sociedad Mexicana de Computación en Educación (SOMECE): <http://www.somece.org.mx>
Syllabus Magazine: <http://www.syllabus.com/article.asp?id=6239>

XML, herramientas de desarrollo y tutoriales

Oxygen XML editor: <http://www.oxygenxml.com>

Procesadores de XSLT, parsers, editores de XML y más en
http://directory.google.com/Top/Computers/Data_Formats/Markup_Languages/XML/Tools
Tutorial de XML de Sun: <http://java.sun.com/xml/learning/tutorial/index.html>
Uso de los XSD: <http://www.codenotes.com/articles/articleAction.aspx?articleID=2093>
XMLSpy: <http://www.altova.com>

Software para evaluación

ExamBuilder es un creador y aplicador de exámenes web: <http://instructor.exambuilder.com>
Evaluador en línea de Prometric Authorized Testing Centers:
<https://www.prometric.com/default.htm>
QuestionMark: <http://www.questionmark.com/esp>
Software educativo gratuito para aplicar exámenes a niños: <http://www.child-reading-tips.com/educational-free-quiz-software-programs.htm#reading>
Shareware y freeware para elaborar y aplicar exámenes:
<http://www.bykeyword.com/pages/exam.html>

Sistemas educativos, ofertas y experiencias de elearning

Blackboard: <http://www.blackboard.com>
Desarrollo de objetos de aprendizaje, tanto de especificaciones como en implementaciones concretas, de Can Studios Ltd.: <http://www.the-can.com>
e42: <http://e42.um.edu.mx>
eduStance: <http://www.edustance.com>
E-learning Institute: <http://www.elearninginstitute.org>
eXact, software para el manejo de todo el ciclo de vida de contenidos para elearning de Giunttilabs: <http://www.giunttilabs.com>
Think Quest: <http://www.thinkquest.org>
WebCT: <http://www.webct.com>
WebQuest: <http://webquest.sdsu.edu>

CD anexo

El CD que acompaña esta obra contiene los siguientes archivos:

- Versión electrónica de la tesis en formato pdf
- Lista de revisión en formato pdf
- Archivos xsd correspondientes a los esquemas XML
- Versión de prueba del XMLSpy 4.0 para 30 días. Activarlo requiere acceder el sitio web de su productor (<http://www.altova.com>) para recibir el código correspondiente.
- Archivo de Rational Rose con los diagramas de UML

REFERENCIAS

- About the World Wide Web Consortium.* (s/f). Recuperado el 2 de abril de 2004 del sitio del World Wide Web Consortium, <http://www.w3.org/Consortium>
- Abran, Alain. (1996). Teaching software engineering using ISO standards. *StandardView*, 4(3), 139-145.
- Advanced Distributed Learning overview.* (2004). Recuperado el 3 de marzo de 2004 de <http://www.adlnet.org/index.cfm?fuseaction=abtadl>
- Alessi, Stephen M y Trollip, Stanley R. (1985). *Computer-Based instruction. Methods and development.* Nueva Jersey: Prentice-Hall.
- Anderson, Jonathan; Durston, Berry H. y Poole, Millicent. (1970). *Redacción de tesis y trabajos escolares.* México: Diana.
- Berge, Zane L. y Collins, Mauri P. (Ed.). (1995). *Computer mediated communication and the online classroom.* Nueva Jersey: Hampton Press.
- Blackboard. *Blackboard Product strategy and vision. White paper on Buildings Blocks Initiative.* (2000). Recuperado el 7 de octubre de 2003 de <http://company.blackboard.com/docs/cp/orientation/BlackboardBuildingBlocks.pdf>
- Bloom, Benjamin S.; Madaus, George F. y Hastings, J. Thomas. (1981). *Evaluation to improve learning.* Nueva York: McGraw-Hill.
- Breitenberg, Maureen A. (1987). *The abc's of standards-related activities in the United States.* Recuperado el 24 de febrero del sitio web de la Office of Standards Code and Information, Office of Product Standards Policy, National Institute of Standards and Technology: <http://ts.nist.gov/ts/htdocs/210/ncsci/stdpnr.htm>
- Burkle, Martha. (2000). *La experiencia de educación en línea en México. Un estudio exploratorio.* Recuperado el 7 de octubre de 2003 de las Memorias del Simposio Internacional de Computación en Educación 2000 de la Sociedad Mexicana de Computación en Educación: <http://www.somece.org.mx/memorias/2000>
- Campbell, Andrea Beth y Pargas, Roy P. (2003). Laptops in the classroom. *Actas del SIGCSE Technical Symposium on computer science education*, 34, 98-102.

- Cardona Ossa, Guillermo. (2002). "Tendencias educativas para el siglo XXI. Educación virtual, online y @learning. Elementos para la discusión". *Eduotec. Revista Electrónica de Tecnología Educativa*, 15. Recuperado el 26 de febrero de 2004 de <http://edutec.rediris.es/Revelec2/revelec15/cardona.pdf>
- Castro Solís, Elizabeth. (2002). *Estándares en los sistemas de gestión de aprendizaje*. Recuperado el 7 de octubre de 2003 de las Memorias del Simposio Internacional de Computación en Educación 2002, de la Sociedad Mexicana de Computación en Educación: <http://www.somece.org.mx/memorias/2002>
- Carretero, Mario y Fairstein, Gabriela. (2001). "La teoría de Piaget y la educación. Medio siglo de debates y aplicaciones". *Pedagogías del siglo XX para el siglo XXI*. Universidad de Buenos Aires. Recuperado el 9 de marzo de 2004 de http://www.mariocarretero.net/spanish/entrevista_piaget.htm
- Collazos O., César Alberto. (2000) . *Internet como nuevo modelo educativo*. Recuperado el 7 de octubre de 2003 de las Memorias del Simposio Internacional de Computación en Educación 2000 de la Sociedad Mexicana de Computación en Educación: <http://www.somece.org.mx/memorias/2000>
- College of Continuing Education. (2002). *Guide to online teaching*. Recuperado el 23 de febrero de 2003 del sitio de la University of Minnesota: <http://ltu.cce.umn.edu/guide>
- Cunningham, George K. (1998). *Assessment in the Classroom: Constructing and interpreting texts*. Londres, The Falmer Press.
- Davis, Alan M. (1983). *Software requeriments. Objects, functions & states*. Nueva Jersey: Prentice-Hall.
- Del Campo Ramírez, Socorro Martín. (s/f). *El papel de la educación artística en el desarrollo integral del educando*. Recuperado el 2 de junio de 2004 de <http://educacion.jalisco.gob.mx/consulta/educar/15/15Martin.html>
- Diario Oficial de la Nación*. (1 de julio de 1992). "Ley federal sobre metrología y normalización". México.
- Díaz Barriga, Ángel (Compilador). (1993). *El examen: textos para su historia y debate*. México, D. F., Universidad Nacional Autónoma de México.
- Dodge, Bernie. (s/f). *Some thoughts about webquests*. Recuperado el 22 de marzo de 2004 del sitio de la San Diego State University: http://edweb.sdsu.edu/courses/edtec596/about_webquests.html
- Glabán Lozano, Sara E. y Ortega Barba, Claudia F. (2002). *Evaluación didáctica de software educativo*. Recuperado el 7 de octubre de 2003 de las Memorias del Simposio

Internacional de Computación en Educación 2002 de la Sociedad Mexicana de Computación en Educación: <http://www.somece.org.mx/memorias/2002>

- Gómez Ferral, Ana Irma; Herrera Ochoa, Esperanza y Bermúdez Lamadrid, Irene. (2001). *Modelo pedagógico de cursos de posgrado a distancia para la formación permanente del profesional de la educación*. Recuperado el 7 de octubre de 2003 de las Memorias del Simposio Internacional de Computación en Educación 2001 de la Sociedad Mexicana de Computación en Educación: <http://www.somece.org.mx/memorias/2001>
- Griffin, John. (2002). *Creación de sitios web con XML y SQL Server 2000*. Madrid: Prentice Hall.
- Hartmann, Jochen; Huang, Shihong y Tilley, Scott. (2001). "Documenting software systems with views II: An integrated approach based on XML". *Actas de la Annual International Conference on Computer Documentation*, 19, 237-246.
- Hermans, H.J.H.; Koper, E.J.R.; Loeffen, A.; Mandelveld, J.M. y Rusman, E.M. (2000). *Reference manual for Edubox-EML/XML binding 1.0/1.0 (beta version)*. Educational Technology Expertise Centre, The Open University of the Netherlands.
- Holzner, Steven. (2001). *Inside XML*. Indiana: New Riders.
- IMS Global Learning Consortium. (2003). Recuperado el 7 de octubre de 2003 de <http://www.imsglobal.org>
- IMS Learning Design Information Model, Version 1.0 Final Specification*. (2003). Recuperado el 7 de octubre de 2003 del sitio del IMS Consortium, <http://www.imsglobal.org>
- Iyangu Pendi, Augusto. (1998). *La educación contemporánea. Teorías e instituciones*. 2da. edición. Valencia, España: AU Llibres.
- Koper, Rob. (2000). *From change to renewal: Educational technology foundation of electronic learning environments*. Educational Technology Expertise Center, The Open University of Netherlands. Recuperado el 21 de marzo de 2004 de <http://learningnetworks.org/forums/showthread.php?s=&threadid=60>
- Koper, Rob. (2001). *Modeling units of study from a pedagogical perspective. The pedagogical metamodel behind EML*. Educational Technology Expertise Center, Open University of Netherlands. Recuperado el 21 de marzo de 2004 de <http://learningnetworks.org/forums/showthread.php?s=&threadid=60>
- L'associació per a la Recerca Pedagògica TP. (Junio de 2003) "Educación y pedagogía". *Boletín*, 20. Recuperado el 2 de junio de 2004 de http://www.llibreriapedagogica.com/butlleti20/educacion_y_pedagogia.htm10.htm
- Lafourcade, Pedro D. (1969). *Evaluación de los aprendizajes*. Buenos Aires: Kapelusz.

- Rada, Roy y Craparo, John. (2000). "Standardizing software projects". *Communications of the ACM*, 42 (12), 21-25.
- Ramírez Sámano, Carlos. (2002). *La formación de docentes en espacios educativos virtuales*. Recuperado el 7 de octubre de 2003 de http://www.cudi.edu.mx/primavera2002/presentaciones/17_30_UPN_carlos.html
- Rehesaar, Hugo. (1996). "International standards: Practical or just theoretical?" *StandardView*, 4(3), 23-127.
- Rice, Gail. (2003). "Designing online tests. The role of assessment in web-based learning". *Journal of Adventist Education*. Abril-Mayo, 31-35.
- Ríos Cabrera, Pablo. (2001). "Concepción del software educativo desde la perspectiva pedagógica". *Quaderns digitales.net*, 24. Recuperado el 26 de febrero de 2004 de www.quadernsdigitals.net/index.php?accionMenu=hemeroteca.VisualizaArticuloIU.visualiza&articulo_id=208
- Rodríguez Illera, José L. (2001). "Diseño y producción de software educativo". *Quaderns digitales.net*, 24. Recuperado el 26 de febrero de 2004 de www.quadernsdigitals.net/index.php?accionMenu=hemeroteca.VisualizaArticuloIU.visualiza&articulo_id=211
- Rosen, Lawrence. (2002). "Geek law: The role of standards in open source". *Linux Journal*. 2002(97), 16.
- Rupf, John A. (2002). "How to add a state-of-the-art component to your course". *Journal of Circuits, Systems, and Computers*, 18(2), 356-361.
- Santángelo, Horacio Néstor. (2000). "Modelos pedagógicos en los sistemas de enseñanza no presencial basados en nuevas tecnologías y redes de comunicación". *Revista Iberoamericana de Educación*, 24.
- Schmidt, Michael E. (2000). *Implementing the IEEE Software Engineering Standars*. Indianapolis: Sams.
- Siebold, Randall J. y Jeffery, Jim. (2003). "Staying mission-centered". *Journal of Adventist Education*. Abril-Mayo, 5-10.
- Shareable Content Object Reference Model*. (2004). Recuperado el 3 de marzo de 2004 de http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci796793,00.html
- Smith, John W. (1997). "Making standard work: A 'whole-product approach'". *ACM Special Interest Group on University and College Computing Services*, XXV, 283-290.

- Software and Information Industry Association. (2003). *Schools Interoperability framework implementation specification. Versión 1.1*. Recuperado el 6 de febrero de 2003 de www.sifinfo.org
- Stacey, Paul. (Febrero de 2003). "People to people not just people to content. Learning environments for active and alive content". Recuperado el 22 de marzo de <http://www.bctechnology.com/statics/pstacey-feb1403.html>
- Stacey, Paul. (Octubre de 2003) "Online pedagogies for active learning". Recuperado el 22 de marzo de 2004 de <http://www.bctechnology.com/statics/pstacey-oct1703.html>
- Stacey, Paul. (Diciembre de 2003). "iPod's, audio blogs, globalization of education & the creative commons. E-Learning christmas wishes from Canada". Recuperado el 22 de marzo de 2004 de <http://www.bctechnology.com/statics/pstacey-dec1903.html>
- Stacey, Paul. (Enero de 2004). "The one percent solution". Recuperado el 22 de marzo de 2004 de <http://www.bctechnology.com/statics/pstacey-jan1604.html>
- Stone, Christopher. (1995). "Software standardization. How the Object Management Group changed the model". *StandardView*, 3(3), 85-89.
- Succi, Giancarlo. (1998). "Compatibility, standards, and software production". *StandardView*, 6(4), 140-146.
- Tattersall, Colin y Koper, Rob. (2003). *EML and IMS Learning Design: From LO to LA*. Educational Technology Expertise Centre, The Open University of the Netherlands. Recuperado el 21 de marzo de 2004 de http://www.ltsn.ac.uk/application.asp?app=resources.asp&process=full_record§ion=generic&id=255
- Tattersall, Colin; Manderveld, Jocelyn; Hummel, Hans; Sloep, Peter; Koper, Rob y de Vries, Fred. (2003). *IMS learning design frequently asked questions*. Educational Technology Expertise Centre, The Open University of the Netherlands. Recuperado de el 21 de marzo de 2004 de <http://learningnetworks.org/forums/showthread.php?s=&threadid=158>
- Tripp, Leonard L. (1996). "International standards on system and software integrity". *StandardView*, 4(3), 146-150.
- Turabian, Kate L. (1996). *A manual for writers of term papers, theses, and dissertations*. Chicago: The University of Chicago Press.
- Turcott, Rosana Verónica. (1999). *Un modelo pedagógico para la integración de medios electrónicos y telemáticos en educación a distancia*. Recuperado el 7 de octubre de 2003 de las Memorias del Simposio Internacional de Computación en Educación 1999 de la Sociedad Mexicana de Computación en Educación: <http://www.somece.org.mx/memorias/1999>

- Updegrave, Andrew. (1995). "Standard setting sand consortium structures". *StandardView*, 3(4), 143-147.
- Ward Jr., William A. (1999). "Some observations on software quality". *Actas de la Annual southeast regional conference*. 34.
- WebCT vs. Blackboard: Report of the Course Management Task Force*. (2002). Recuperado el 2 de diciembre de 2003 del sitio web de la Western Carolina University: <http://www.wcu.edu/it/cio/planning/CMSfinalreport.pdf>
- WebQuest Taskonomy: A Taxonomy of Tasks*. Recuperado el 22 de marzo de 2004 de <http://webquest.sdsu.edu/taskonomy.html>
- White, Elena G. (1987). *La educación*. Bogotá: Asociación Publicadora Interamericana.
- Wiggins, Grant P. (1998). *Educative assessment. Designing assessments to inform and improve student performance*. San Francisco: Jossey-Bass.
- Wiggins, Grant. (1999). *The case for authentic assessment*. Recuperado el 22 de marzo de 2004 de <http://pareonline.net/getvn.asp?v=2&n=2>
- Woit, Denise y Mason, David. (2003). "Effectiveness of online assessment". *Actas de la SIGCSE technical symposium on Computer science education*, 34.
- Zavala Ruiz, Jesús María. (2002). *Ingeniería de Software*. Recuperado el 4 de abril de 2004 de <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>
- Zorrilla, Santiago y Torres Xammar, Miguel. (1986). *Guía para elaborar la tesis*. México: McGraw-Hill.