

SOPORTE EN PLATAFORMAS DE E-LEARNING A PATRONES PEDAGÓGICOS

JESÚS CRISTÓBAL BARRIOS

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin Máster en Ingeniería de Computadores / Ingeniería Informática para la Industria /
Programación y Tecnología Software / Sistemas Inteligentes

07 de septiembre de 2011

Director: Antonio Navarro Martín
Colaborador externo: Félix Buendía García (UPV)

Autorización de Difusión

JESÚS CRISTÓBAL BARRIOS

07 de septiembre de 2011

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “SOPORTE EN PLATAFORMAS DE E-LEARNING A PATRONES PEDAGÓGICOS”, realizado durante el curso académico 2010-2011 bajo la dirección de Antonio Navarro y con la colaboración externa de dirección de Félix Buendía (Universidad Politécnica de Valencia) en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Resumen en castellano

El e-learning ha tenido un creciente éxito en los últimos años. Cada vez más instituciones educativas incorporan plataformas de e-learning en sus procesos docentes. Sin embargo, los docentes no suelen utilizar un método sistemático para aprovechar la potencia de estas plataformas a la hora de aplicar patrones pedagógicos. Los patrones pedagógicos son soluciones que pueden ser reutilizadas por los docentes para plasmar sus necesidades educativas en las plataformas e-learning y constituyen, además, una manera excelente de transmitir experiencias compartiendo un lenguaje común. Este trabajo revisa los principales mecanismos de descripción de patrones de pedagógicos, proponiendo un proceso para la adecuación y evaluación de estos patrones dentro de un ambiente de e-learning.

Palabras clave

Learning Management System, Patrones e-learning, Patrones pedagógicos, contenidos educativos, escenarios de aprendizaje, gestión del conocimiento.

Resumen en inglés

E-learning has been increasingly successful in recent years. Thus, every day more educational institutions incorporate e-learning platforms in their educational processes. However, teachers usually do not use systematic methods to take advantage of the power of these platforms when applying pedagogical patterns. Pedagogical patterns are solutions that can be reused by teachers to shape their educational needs in e-learning platforms and are also an excellent way to convey experiences sharing a common language. This paper reviews different pedagogical patterns, proposing a process for the adaptation and evaluation of these patterns within an environment of e-learning.

Keywords

Learning Management System, e-learning patterns, educational content, pedagogical patterns, learning, knowledge management scenarios.

Índice de contenidos

Autorización de Difusión.....	ii
Resumen en castellano.....	iii
Palabras clave.....	iii
Resumen en inglés	iv
Keywords	iv
Índice de contenidos	1
Agradecimientos	2
1. Introducción.....	5
2. Patrones y catálogo.....	9
3. Funciones de un LMS genérico	25
I. Herramientas orientadas al aprendizaje	25
II. Herramientas orientadas a la productividad.....	29
III. Herramientas para la interacción/participación de los estudiantes.	31
IV. Herramientas de soporte.....	32
V. Herramientas destinadas a la publicación de cursos y contenidos.....	33
VI. Herramientas para el diseño de planes de estudio.....	34
VII. Sistemas para la gestión del conocimiento en el ámbito educativo	35
4. Propuesta de funciones para patrones.....	37
5. Validación.....	70
6. Conclusiones y trabajo futuro.....	81
Bibliografía	84

Agradecimientos

“Mi trabajo, como profesor no es enseñar a mis alumnos lo que sé, ya que gran parte no será de utilidad, especialmente en un campo tan cambiante como la informática. Mi trabajo, creo yo, es enseñarles cómo pienso yo. No es algo que deban imitar en cualquier caso, pero al menos a mí me ha servido para seguir los cambios, retos y oportunidades”.

Bergin

Acrónimos

AACV	-	Arquitecturas Avanzadas en Campus Virtuales
ADL	-	Advanced Distributed Learning Initiative
AICC	-	Aviation Industry CBT Committee
AWT	-	X-Window Toolkit
CAL	-	Computer-Aided Learning
CBT	-	Computer-Based Training
CMS	-	Content Management System
DAV	-	Distributed Authoring and Versioning
EAO	-	Enseñanza Asistida por Ordenador.
ED	-	Educación a Distancia.
EML	-	Educational Modeling Language
EVA	-	Entorno Virtual de Aprendizaje
FAQ	-	Frequently Asked Questions
HTML	-	HyperText Markup Language
HTTP	-	HyperText Transfer Protocol
IEEE	-	Institute of Electrical and Electronic Engineers
IMS	-	IMS Global Learning Consortium
IP	-	Internet Protocol
ISO	-	International Organization for Standards
ITS	-	Intelligent Tutoring System
LDAP	-	Lightweight Directory Access Protocol
LCMS	-	Learning Content Management System
LMS	-	Learning Management System
LOM	-	Learning Object Metadata
LP	-	Learning Platform
MDA	-	Model-Driven Architecture
PC	-	Personal Computer
PCeL	-	Person Center e-Learning
QTI	-	Question and Test Interoperability
RDF	-	Resource Description Framework
RLO	-	Reusable Learning Objects
RSS	-	Really Simple Syndication
SCORM	-	Shareable Content Object Reference Model
VoIP	-	Voice over IP
WBL	-	Web-Based Learning
WBT	-	Web-Based Training
W3C	-	World Wide Web Consortium

1. Introducción

El uso de las *Tecnologías de la Información y la Comunicación* (TIC) en educación ha beneficiado a la comunidad educativa durante años [1]. En la actualidad dicho uso se denomina *e-learning* [2], y se ha convertido en un factor clave para el éxito de las universidades [1, 3, 4, 5, 6, 7]. Los Campus Virtuales constituyen la apuesta institucional en los centros de enseñanza superior por el e-learning. Dichos campus virtuales están destinados a funcionar como espacios para la enseñanza, aprendizaje e investigación, creados mediante la confluencia de múltiples aplicaciones de las TIC: Internet, Web, comunicación electrónica, vídeo, vídeo-conferencia, multimedia y publicación electrónica [80] El uso de estos campus virtuales ha crecido enormemente en los últimos años. En 2001/02 apenas 36 de las 68 universidades españolas contaban con algún campus virtual (en exclusiva o compartido), sin embargo, en el curso 2007/08, prácticamente el total de las 74 instituciones de educación superior españolas (de 5 se desconocía la información) contaban con algún campus virtual [85].

Sin embargo, a pesar de la continua expansión del e-learning, no son pocos los autores que hablan de un supuesto fracaso como opción formativa [8, 9]. Estos autores apuntan que las causas de este “fracaso” se centran en no haber cubierto las expectativas que en este modelo había depositadas. Podríamos citar otro tipo de causas que justifiquen este fracaso. Entre otras, caben destacar: el uso de tutores “low cost” que delegan demasiado en el modelo e-learning la tarea educativa; la sobrecarga de trabajo, pues muchos de los actores de los sistemas e-learning lo son a la vez de los sistemas de enseñanza tradicionales; el excesivo peso de aprendizaje en materiales que no requerían intervención humana; la presencia de entornos virtuales de enseñanza excesivamente rígidos demasiado preocupados del marketing; el hecho que en ambientes universitarios el docente deba dedicar más tiempo a su docencia virtual en detrimento de su labor investigadora, etc. Además, hay que recordar que, hasta el momento, más de un 80% de fracaso en cursos gestionados a distancia y más de un 60% de abandono de los mismos por parte de los estudiantes [10].

El modelo e-learning ha favorecido el uso de materiales digitales y ha modificado la forma de transmitir el conocimiento. De hecho, hoy es difícil entender un proceso de enseñanza que no haga uso más o menos intensivo de las Tecnologías de la Información y la Comunicación, pues ya no se trata de aprender más o mejor, sino de aprender diferente.

Cabero [11] recoge en la Tabla 1 algunas diferencias entre la formación presencial y la basada en el uso del e-learning en las aulas.

Formación basada en la red	Formación presencial tradicional
<ul style="list-style-type: none"> - Permite que los estudiantes vayan a su propio ritmo de aprendizaje. - Es una formación basada en el concepto de <i>formación en el momento en que se necesita (just-in-time training)</i>. - Permite la combinación de diferentes materiales (auditivos, visuales y audiovisuales). - Con una sola aplicación puede atenderse un mayor número de estudiantes. - El conocimiento es un proceso activo de construcción. - Tiende a reducir el tiempo de formación de las personas - Tiende a ser interactiva, tanto entre los participantes en el proceso (profesores y estudiantes) como con el contenido - Tiende a realizarse de forma individual, sin que ello signifique la renuncia a la realización de propuestas colaborativas - Puede utilizarse en el lugar de trabajo y en el tiempo disponible por parte del estudiante - Es flexible - Tenemos poca experiencia en su uso - No siempre disponemos de los recursos estructurales y organizativos para su puesta en funcionamiento. 	<ul style="list-style-type: none"> - Parte de una base de conocimiento, y el estudiante debe ajustarse a ella. - Los profesores determinan cuándo y cómo los estudiantes recibirán los materiales formativos - Parte de la base de que el sujeto recibe pasivamente el conocimiento para generar actitudes innovadoras, críticas e investigadoras. - Tiende a apoyarse en materiales impresos y en el profesor como fuente de presentación y estructuración de la información. - Tiende a un modelo lineal de comunicación. - La comunicación se desarrolla básicamente entre el profesor y el estudiante. - La enseñanza se desarrolla de forma preferentemente grupal. - Puede prepararse para desarrollarse en un tiempo fijo y en aulas específicas. - Se desarrolla en un tiempo y en aula específicas. - Tiende a la rigidez temporal - Tenemos mucha experiencia en su utilización. - Disponemos en muchos recursos estructurales y organizativos para su puesta en funcionamiento

Tabla 1. Características de la formación presencial y en red [11].

A menudo se nos propone el uso de distintos *Learning Management Systems* (LMS) o plataformas de e-learning (por ejemplo Sakai [12], Moodle [13], Blackboard [14]), para crear Entornos Virtuales de Aprendizaje (EVAs) que den solución a este problema. Sin embargo, en muchas ocasiones, el uso de dichos entornos suele complicar el proceso. Para que un EVA tenga éxito es preciso afrontar una serie de retos tanto pedagógicos como tecnológicos que hagan que el curso final facilite realmente el proceso de aprendizaje. Existen retos pedagógicos que derivan de la necesidad de convertir estos sistemas, no sólo en meros contenedores de información digital, sino que ésta debe ser transmitida de acuerdo a unos modelos y patrones pedagógicamente definidos para dar satisfacción a las necesidades específicas de los estudiantes.

De igual forma, existen retos a medio camino entre la pedagogía y la tecnología. Por ejemplo, un diseño de la interfaz poco adecuado, o estructuras de menú poco intuitivas que distraigan al profesor/estudiante de su objetivo de enseñanza/aprendizaje, perdiendo el tiempo en aprender cómo se

usa el sistema en vez de centrarse en enseñar/aprender un concepto o una habilidad, puede incidir en el fracaso del sistema, a pesar de contar con contenidos de alta calidad. También nos encontramos con otra importante barrera. No se aprende y se enseña igual en un entorno virtual que en un entorno presencial. Por lo tanto, es necesario capacitar a los tutores para el entorno virtual, formarles en los nuevos modelos pedagógicos, en la adaptación a la diversidad y métodos de aprendizaje activo (aprendizaje cooperativo y colaborativo, resolución de problemas, simulaciones, aprendizaje basado en casos, aprendizaje basado en proyectos, etc.).

Se hace por tanto necesario crear soluciones específicas en las áreas de la pedagogía y la informática para resolver los problemas que plantea el e-learning.

Si observamos el comportamiento de los expertos, sea cual sea su campo de especialización, no suelen crear soluciones completamente nuevas en cada problema que se presenta, sino que se basan en su experiencia para adaptar soluciones que ya han funcionado en la resolución de problemas anteriores. Así pues, suelen reutilizar su conocimiento para aplicarlo en los nuevos problemas [46]. En muchos entornos, a este conocimiento reutilizado se le suele denominar *patrón*.

Este trabajo revisa algunos de los principales patrones relacionados con el uso de sistemas e-learning desde aquellos puramente pedagógicos (por ejemplo, patrones cuya aplicación requiere la presencia de los alumnos en el aula) hasta aquellos más centrados en el uso de estos patrones en un entorno virtual (por ejemplo, la organización de los alumnos en grupos de trabajo colaborativos con un objetivo común). En particular, el trabajo se centra en los patrones pedagógicos identificados por Bergin [34]. El objetivo es doble: (i) validar la aplicabilidad de estos patrones en entornos de e-learning; y (ii) proponer y validar el uso de herramientas pertenecientes a LMSs con el fin de dar soporte a la aplicación de los patrones de Bergin en el e-learning.

Cabe destacar que las aportaciones de este trabajo son de especial relevancia para el proyecto *Arquitecturas Avanzadas en Campus Virtuales*, AACV (TIN2009-14317-C03-01), que busca construir campus virtuales basados en una arquitectura software orientada a servicios [18], y que dan respuesta ágilmente a las necesidades cambiantes de sus usuarios mediante una arquitectura dirigida por modelos [19]. Así, en la medida que este trabajo determine las funciones principales que deben estar en toda plataforma e-learning que de soporte a un campus virtual, y utilice dichas funciones de manera óptima para dar un soporte informático idóneo a dichos patrones, los campus virtuales construidos serán capaces de cumplir mejor los requisitos impuestos por sus usuarios.

De esta forma, el Capítulo 2 lo dedicamos a repasar distintos conceptos relacionados con el uso de los patrones en el ambiente de la enseñanza, no necesariamente asistida por las TIC, realizando una exploración en algunos de los patrones más ampliamente utilizados. En el Capítulo 3 se describen las herramientas que deben caracterizar un LMS genérico de manera que pueda servir de apoyo en las tareas educativas. En el Capítulo 4 mostramos un proceso de adaptación de patrones pedagógicos a la utilización de las herramientas mostradas en el capítulo anterior. En el Capítulo 5 se ofrecen los resultados de validación de la aplicación propuesta en el capítulo anterior. Finalmente se presentan las conclusiones en la aplicación de estos patrones y el trabajo futuro.

2. Patrones y catálogo

Educación a distancia

Para encontrar las primeras experiencias en la educación a distancia, seguramente deberíamos remitirnos al origen de la humanidad donde el ser humano transmitió experiencias y conocimiento mediante pinturas rupestres, utensilios, etc. Sin embargo, históricamente, la Educación a Distancia (ED) empezó entre Europa y Estados Unidos a finales del siglo XIX empleando los sistemas de correspondencia postal tradicionales. Estas primeras experiencias comenzaron a través de emigrantes del norte de Europa que se encontraban en EEUU y deseaban que sus hijos fueran formados por docentes que estaban en sus países de origen con la finalidad de mantener vivas sus costumbres y culturas locales. Garrison [20,21] identifica tres fases en la historia de la Educación a Distancia.

La primera fase caracterizada por la *enseñanza por correspondencia* (finales del siglo XIX y principios del XX) se basaba en textos muy rudimentarios y poco adecuados para el estudio independiente de los alumnos. El sistema de transmisión eran, básicamente, los servicios postales, muy fiables pero también muy lentos, con lo que la interacción entre profesores y alumnos se encontraba demasiado limitada. Metodológicamente, no existía ninguna especificidad didáctica, se trataba de reproducir por escrito una clase presencial tradicional. Poco se tardó en apreciar que así el aprendizaje no era fácil [22], por lo que se trató de dar una forma más interactiva a ese material escrito mediante el acompañamiento de guías de ayuda al estudio, actividades complementarias, cuadernos de trabajo ejercicios de evaluación. A partir de mediados del siglo XIX aparecen distintos inventos que facilitan la comunicación entre el formador y los alumnos: telégrafo (1830), teléfono (1876), radio (1894) y televisión (1923). Esta etapa ha sido la de mayor duración. Si analizamos hoy multitud de realizaciones de educación a distancia en el mundo, observaremos que, en el siglo XX, muchas de ellas no habían traspasado aún esta primera generación, siendo en buena parte de los países la forma más popular de desarrollar esta enseñanza [23].

Una segunda etapa la constituye la denominada *etapa multimedia* y que podría situarse a finales de los años setenta (creación de la *Open University* británica) donde los medios de masas como radio y televisión, presentes en la mayoría de los hogares, facilitan la integración de materiales audiovisuales en el proceso de aprendizaje. También el teléfono, vídeos, audio casetes, etc., enriquecen la calidad del material que llega hasta el alumno. Sin embargo el sentido de la comunicación sigue siendo básicamente unidireccional.

La tercera y, hasta ahora, última etapa, es la denominada *etapa telemática* que comenzaría a aplicarse en la década de los 80. La integración de las telecomunicaciones con otros medios educativos, mediante la informática define a esta etapa. Esta tercera generación se apoya en el uso cada vez más generalizado del ordenador personal y de las acciones realizadas en programas flexibles de Enseñanza Asistida por Ordenador (EAO) (*Computer-Based Learning, CBL o CBT*) [82] y de sistemas multimedia.

Desde que en 1965 Skinner [24] propusiera el concepto de “máquinas de enseñar”, el desarrollo y preocupación de la utilización de los ordenadores en los procesos de enseñanza se ha centrado en responder a esta pregunta: ¿de qué manera el ser humano puede aprender a través de la interacción, casi exclusiva, con una máquina?. Desde un principio los ensayos fueron encaminados a construir objetos físicos que, sin la intervención directa de un tutor o profesor, propiciara el aprendizaje (*conductivismo*) [78]. Sin embargo, en la actualidad, con los avances en el campo de la multimedia, y sobre todo de las telecomunicaciones y de internet, están cobrando auge los proyectos y métodos educativos basados en las tesis *socioconstructivistas* del aprendizaje [25] donde el individuo moldea su aprendizaje basado en el descubrimiento y la interacción con otros individuos. Ambas corrientes ofrecen diferentes ópticas sobre como el individuo aprende, pero no constituyen una metodología o teoría sobre la enseñanza. De hecho, es correcto considerar que ambas concepciones ofrecen ejemplos exitosos en su aplicación en el proceso de enseñanza-aprendizaje [79].

La separación entre ambas teorías no aparece claramente delimitada y así es posible identificar corrientes paralelas a medio camino entre ambas concepciones. A modo de resumen, se ofrece en la Tabla 2 un cuadro donde se sintetizan las diferencias entre ambos enfoques pedagógicos que tratan de entender cómo se produce el aprendizaje en el individuo.

	Conductismo	Constructivismo
Supuestos epistemológicos	Objetivismo, realismo, empirismo.	Pragmatismo, racionalismo.
Objeto de estudio	La conducta.	La construcción del conocimiento.
Principales autores	E.L. Thorndike, John B. Watson, B.F. Skinner.	Jean Piaget, Lev Vigotsky, Jerome Bruner, Howard Gardne
Definición de aprendizaje	Cambio en la probabilidad de ocurrencia de una conducta en un contexto dado.	Cambio en los significados, contruidos a partir de la experiencia.

	Conductismo	Constructivismo
Descripción del proceso de aprendizaje	El aprendizaje se produce por medio de los estímulos antecedentes y las consecuencias de las conductas (estímulos externos al organismo).	El aprendizaje se da a partir de la interacción entre el conocimiento previo, el contexto social y el problema por resolver.
Conceptos principales	Conducta (condicionamiento) operante, reforzamiento, castigo, extinción, programas de reforzamiento, moldeamiento de la conducta, generalización.	Asimilación y acomodación, aprendizaje significativo, estilos de aprendizaje, inteligencias múltiples, andamiaje, construcción social de conocimientos, aprendizaje colaborativo.
Métodos de enseñanza asistida	Enseñanza asistida por ordenador (EAO), multimedia educativo en CD-ROM, cursos empaquetados on-line.	Proyecto LOGO, videojuegos simulación, Webquest, círculos aprendizaje.
Características	Material de enseñanza estructurado. Aprendizaje por recepción. Se aprende como actividad individual del alumno con el ordenador.	Material organizado en torno a problemas y actividades. Aprendizaje por descubrimiento. Relevancia del trabajo colaborativo.

Tabla 2. Principales teorías sobre el aprendizaje.

Las primeras experiencias en el campo de la Enseñanza Asistida por Ordenador se centraban en cuatro procesos secuenciales idénticos para todos los alumnos: 1) el profesor ponía distintos materiales a disposición del alumno. 2) el alumno recibía este material y lo asimilaba. 3) el alumno era examinado del conocimiento adquirido a través de estos materiales, y 4) si el alumno superaba estos exámenes, el profesor ponía a disposición del alumno nuevos materiales y comenzaba de nuevo el ciclo (conductismo [78]). Las ventajas de estos procesos son evidentes (ver Tabla 1): permitían al alumno de todo el material necesario para aprender un tema, incluidos ejercicios y simuladores y, además, cada alumno podía elegir sus momentos y ritmo de aprendizaje. Estos materiales podían ser ofrecidos a un número de alumnos mucho mayor y además se abaratarían los costes destinados al formador, sustituyendo a este por un comunicador o facilitador. Las desventajas, también son claras: no todos los alumnos asimilan el mismo material de la misma manera, pues no todos tienen las mismas destrezas y conocimientos, y dada la poca interactividad con el formador, el alumno puede sentirse “descolgado” del resto de sus compañeros. Además no existe mucho material de calidad adaptado a las diversas situaciones y temas de formación para que el alumno pueda aprender de forma “aislada”, sin olvidar la oposición del alumno a aprender

dirigido por “una máquina”. Frente a esta propuesta, S.Papert desarrolló un proyecto (LOGO¹) - de enseñanza basado en ordenadores apoyado en el supuesto de que el conocimiento es básicamente producto de una experiencia reconstruida por los sujetos y, por ello, el ordenador más que una máquina que enseña transmitiendo información se convierte en un instrumento o recurso para adquirir experiencias potencialmente educativas (constructivismo [79]).

Otro enfoque de gran importancia en la enseñanza con ordenador es el denominado Sistema Tutor Inteligente (*Intelligent Tutoring System, ITS*) [83] que pretende generar programas educativos basados en los principios de la inteligencia artificial y cuya evolución posterior llevo a formular el concepto de sistemas expertos educativos. Estos sistemas se entienden como la capacidad de toma de decisiones de un aplicativo en función de una experiencia con el propósito de emular un comportamiento parecido al de un tutor humano. Se caracterizan por la flexibilidad y adaptación del sistema al comportamiento mostrado por el alumno en vez de responder rígidamente a un patrón previamente establecido.

Desde el punto de vista tecnológico, cualquiera que sea el escenario elegido para cubrir los procesos de educación a distancia, y más concretamente, de la enseñanza asistida por ordenador, deberá contemplar una serie de herramientas que cubran desde la gestión de recursos electrónicos , hasta los procesos de aprendizaje (p.e. Colaboración entre alumnos, tareas, test, etc.).

Conceptos básicos

A continuación se detallan algunos conceptos básicos utilizados lo largo de este trabajo, que hacen referencia a términos comúnmente empleados en el mundo del e-learning.

Una *Plataforma de Aprendizaje* (LP) o EVA es el conjunto de herramientas y software destinado a organizar y facilitar el aprendizaje basado en entornos Web (WBL – *Web-Based Learning*). La mayoría de los LP se centran en una arquitectura cliente/servidor, donde el servidor del LP recibe y procesa las peticiones solicitadas desde los clientes. El cliente puede utilizar cualquier tipo de software para interactuar con el servidor, aunque son los navegadores web (IE Explorer, Chrome, Mozilla, etc.) los más utilizados. No obstante existen otras soluciones para implementar una LP, sin ser necesariamente WBL:

- Aplicaciones Monolíticas, donde es necesario instalar un software propio en el PC. Corresponde a este segmento aplicaciones multimedia instaladas a partir de un CD-

¹ <http://el.media.mit.edu/logo-foundation/>

ROM que permiten interactuar con el alumno sin necesidad de una conexión a internet. Pueden llegar a ser muy completas y con claras ventajas en cuanto a rendimiento.

- Aplicaciones Distribuidas, a medio camino entre las aplicaciones monolíticas y la arquitectura cliente/servidor. En ellas, el software necesario para gestionar el contenido multimedia reside en el propio equipo, pero se habilita la conexión por internet a otros equipos para completar ciertos procesos de enseñanza (p.e.: Comunicación entre alumnos, información sobre la evolución del alumno, etc.).

A pesar de esta división de LP, los más ampliamente utilizados (vendidos) son aquellos que utilizan entornos Web con estándar HTTP y HTML [26], donde tan solo es necesaria la utilización de un navegador y algunas herramientas (plug-ins) autoinstalables para visualizar ciertos contenidos multimedia (p.e.: Animaciones 3D, reproductores Flash, reproductores de vídeo y audio, etc.). A este segmento también se le denomina LMS (*Learning Management System*).

Si consideramos la evolución de los LMS desde su nacimiento [91], debemos retroceder a los años 1960, con el nacimiento del sistema PLATO² (Programmed Logic for Automated Teaching Operation), desarrollado por la Universidad de Illinois como un sistema monopuesto ejecutado sobre la computadora ILLIAC³. Este sistema funcionó durante cuatro décadas llegando a tener más usuarios, en sus primeros diez años, que ARPANET [90], precursora de internet. Desde este momento hasta nuestros días la aparición de plataformas LMS se ha multiplicado ofreciendo nuevas funcionalidades y mayor integración con sistemas de gestión. Estas plataformas van desde las puramente comerciales como Sharepoint, Desire2learn, Blackboard o Authorware, a las de código abierto, como Moodle, Sakai, dotLRN, Claroline o Dokeos, pasando por las de diseño propio dentro de la propia institución [57,58,59,60,61,62,63,64,65,66].

Junto con el desarrollo de los LMS surge el crecimiento y progreso de los CMS (Content Management System), que son aplicaciones software destinadas a crear y gestionar información multimedia online para ofrecerla mediante páginas web dinámicas, de forma que se sepan los contenidos de la información. Por tanto, no se centra tanto en el progreso del usuario sino en el despliegue de los

² <http://platohistory.org/>

³ http://en.wikipedia.org/wiki/ILLIAC_I

contenidos. La combinación de ambos enfoques es lo que se ha denominado LCMS (Learning Content Management System).

Con frecuencia se confunden los LMS con los LCMS cuya principal ventaja es gestionar objetos de aprendizaje que puedan convertirse en objetos reutilizables (RLO – Reusable Learning Objects) y de esta forma, poder ser empleados entre distintos LMS para lo cual deben seguir ciertos estándares [92]. Los contenidos usualmente se almacenan como objetos descritos e identificables de forma única. En un LCMS se tienen contenedores o repositorios para almacenar los recursos, que pueden ser utilizados de manera independiente o directamente asociados a la creación de cursos dentro del mismo sistema. Es decir, que el repositorio puede estar disponible para que los profesores organicen los cursos o también pueden estar abiertos para que cualquier usuario recupere recursos no vinculados a ningún curso en particular, pero que les pueden ser de utilidad para reforzar los aprendidos sobre algún tema.

En resumen, un LCMS es un sistema basado en web que es utilizado para crear, aprobar, publicar, administrar y almacenar recursos educativos y cursos en línea [27].

Como conclusión, se puede afirmar que tanto los LMS como los LCMS se pueden generalizar como sistemas de gestión de aprendizaje ya que los primeros gestionan la parte administrativa de los cursos, así como el seguimiento de actividades y avance del alumno; mientras que los segundos gestionan el desarrollo de contenidos, su acceso y almacenamiento. En el mercado, los más comunes son los LMS ya que la complejidad de los LCMS los ha llevado a un desarrollo más lento [28].

La adopción de un LMS por constituye una decisión importante por parte de la institución, pues de ello depende gran parte de las funcionalidades que se ofertarán a profesores y alumnos, sin embargo no supone más que un paso dentro de la implantación de un Campus Virtual dentro de un ambiente universitario. Según el análisis de Khan [29], en un proceso de e-learning deben tenerse en cuenta ocho dimensiones diferentes: diseño institucional, pedagógico, tecnológico, del interfaz, evaluación, gestión, soporte y ética de uso [Figura 1]. Este análisis confirma que el e-learning no trata solamente de tomar un curso y colocarlo en un ordenador, se trata de combinar recursos, interactividad, apoyo y actividades de aprendizaje estructuradas.

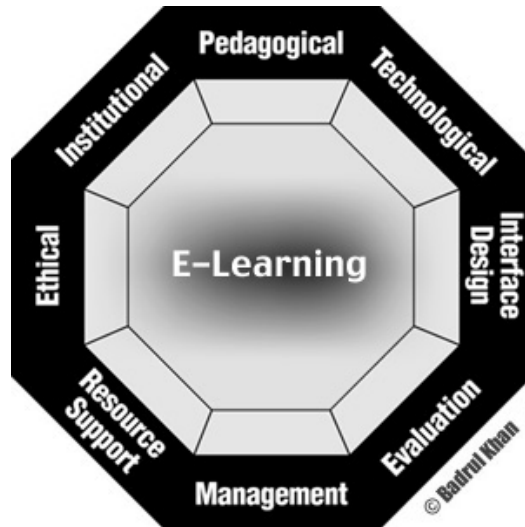


Figura 1. Entorno e-learning.

Patrones

El concepto de patrón de diseño surge en el campo de la arquitectura civil de la mano del arquitecto Christopher Alexander [15] como una manera de documentar la esencia de una solución a un problema recurrente en el diseño arquitectónico. Alexander detalla una colección de 253 patrones dentro del ámbito de la arquitectura y el urbanismo, destinados a mostrar prácticas habituales en diseño dependiendo de la región, ciudades, tipo de construcción, etc. Dos décadas después, la idea de Alexander fue acogida con rotundo éxito dentro de la ingeniería del software tras la publicación del libro *Design Patterns: Elements of Reusable Object-Oriented Software* [16]. Este libro describía 23 patrones utilizados en el diseño de sistemas de software y sirvió para asentar las bases de los patrones de diseño en la comunidad del desarrollo de software orientado a objetos y convertir los patrones de diseño en un tema central de la reutilización software.

En general, podemos decir que una determinada respuesta, actuación o procedimiento alcanza el estatus de patrón cuando ha demostrado en un número suficiente de veces su validez. Es decir, cuando ha proporcionado una solución correcta y, además, es una de las mejores soluciones desde el punto de vista de la economía de medios [17].

Por lo general, los patrones nunca se utilizan de manera aislada sino en combinación con otros patrones, con el fin de resolver problemas complejos. Surgen como una técnica para compartir y capturar conocimiento sobre un determinado ámbito de diseño.

La utilización de patrones no conlleva que todo vaya a salir bien, sólo muestra una colección de buenas prácticas que permiten al usuario experimentar con pautas probadas en otros entornos similares al suyo, anteponiéndose a problemas que puedan surgir de la utilización de estos patrones y conociendo los problemas o consecuencias que puede suponer la aplicación de estos patrones. De hecho, Alexander [48] describe al patrón como una regla de tres partes, que expresa una relación entre un contexto, un problema y una solución. Por tanto, debemos ver los patrones, no como una solución en sí misma, sino como la documentación de la forma en que se afrontaron problemas similares en el pasado dentro de un contexto determinado.

En general, cualquier catálogo de patrones utiliza un lenguaje estructurado que homogeniza, por un lado el vocabulario empleado y, por otro, la información que de cada patrón se precisa. Describir un grupo de patrones puede servir de ayuda a muchos profesionales, pero esto último será difícil si el usuario es incapaz de identificar para qué sirve cada patrón, cómo y cuándo utilizarlo, ni siquiera si el problema que resuelve tiene relación con el suyo. Por este motivo, tan importante es elaborar un patrón como documentarlo adecuadamente.

Una descripción de un patrón debe servir para algo más que para crear un catálogo de patrones [49]. Debe indicar que problema resuelve, cuando aplicar estos patrones, como aplicarlos, el resultado de aplicarlos, etc. No existe aún ninguna estandarización sobre cómo abordar el desarrollo de estas descripciones, ni ninguna clasificación que las relacione, si bien todas pretenden documentar los patrones mediante fichas con distinta estructura de campos. En este proyecto utilizaremos los catálogos de patrones sin ahondar en la estructura utilizada para documentar dichos patrones, ya que dicha estructura no es el objetivo de este trabajo. El lector interesado puede ampliar información en [48].

Documentar buenos patrones puede ser una tarea muy difícil. Coplien [50] indica que un buen patrón:

1. Resuelve un problema: Los patrones aportan soluciones, no principios o estrategias abstractas.
2. Es un concepto probado: Proponen soluciones, no teorías o especulaciones. En el caso del “*Design Patterns*”, el criterio para decidir si algo era un patrón o no, era que éste debía tener al menos 3 implementaciones reales.
3. La solución no es obvia: Muchas técnicas de resolución de problemas (como los paradigmas o métodos de diseño de software) intentan derivar soluciones desde principios básicos. Los mejores

patrones generan una solución a un problema indirectamente (un enfoque necesario para los problemas de diseño más difíciles).

4. Describe una relación: Los patrones no describen módulos sino estructuras y mecanismos.
5. Tiene un componente humano significativo: La aplicación de un mismo patrón puede proporcionar soluciones distintas dependiendo del despliegue.

Pueden verse patrones en muchas áreas del saber: arquitectura, matemáticas, ecología [72], geología [74], sicología [71,73], biología [75], ingeniería de software, pero es difícil saber cuándo una disciplina puede beneficiarse de su uso. Incluso dentro de campo de la informática, continúa aflorando el uso de patrones en distintas áreas:

- Web (Hipermedia Design Pattern) [55,56,70]
- Arquitectura del software [68, 69]
- Análisis orientado a objetos[86]
- Modelo de Negocio[87]
- Interacción persona-ordenador [84, 88]
 - Interface de usuario
 - Diseño Web
 - Usabilidad
 - Seguridad
 - etc.

Problemas y limitaciones en el uso de patrones

El uso de patrones puede parecer la solución universal a todos nuestros problemas, sin embargo, no es así. No se trata de convertir a los patrones en la *bala de plata* que solucione todas nuestras dificultades. Existen diseños maravillosos donde no se ha utilizado ningún patrón (al menos de forma consciente) y diseños donde se han aplicado patrones a “rajatabla” y donde el funcionamiento es, cuando menos, de dudosa efectividad. Entonces, ¿Cuándo debemos utilizar patrones y cuándo no?

Como hemos indicado anteriormente, los patrones constituyen una forma práctica de documentar experiencias de otros autores, pero eso no necesariamente implica que estas experiencias sean efectivas en nuestro propio entorno, es decir, las experiencias transmitidas pueden constituir *falsos patrones* que no

reflejen el consenso de varios profesionales o que no documenten de forma adecuada el contexto en donde deben aplicarse.

La documentación de los patrones debe ser lo suficientemente abstracta como para contemplar la heterogeneidad de su uso, pero a la vez debe ser lo suficientemente clara como para facilitar su aplicabilidad, es decir, la documentación de los patrones debe utilizar una *estructura común* (lenguaje de patrones) que muestre cada patrón como una solución a un *problema identificable* por el propio usuario. Esta puede ser la diferencia entre patrones suficientemente maduros y patrones pobremente probados y documentados.

A cada momento aparecen nuevos patrones para problemas específicos. Por ello, habrá que vigilar que el catálogo de patrones no sea excesivamente *disperso*, pues, por un lado, dificultan la tarea de identificar qué problema resuelven y, por otro lado, hace más difícil la interacción entre los distintos patrones.

Otro problema muy común lo constituye el exceso en la utilización de patrones, lo que se denomina *patterns happy* [89]. A veces los usuarios aprenden un patrón e inmediatamente lo aplican sin que su utilización aporte ningún valor añadido, esto no hace más que complicar el escenario. De igual manera que elaboramos la lista de la compra antes de acudir a unos grandes almacenes para no terminar comprando lo que no se necesita, el usuario debe conocer los problemas que debe resolver antes aplicar de forma indiscriminada patrones para resolver supuestos problemas. Así, por ejemplo, durante el diseño del Campus Virtual de la UCM, no se aplicaron todos los patrones del catálogo de Alur [98], sino aquellos que pensamos podían ser más útiles en nuestro diseño [99,100]. De hecho, no sólo el diseño de un Campus Virtual es complejo. Incluso la infraestructura de ingeniería software que lo sustenta es de una cierta complejidad [101].

Otro aspecto importante a tener en cuenta a la hora de aplicar patrones, es que los catálogos de patrones muestran experiencias exitosas para solucionar problemas, pero no dice nada de las *experiencias desastrosas* que los usuarios aplican una y otra vez de forma intuitiva sin que nadie les alerte de los problemas que pueden encontrar. Estamos hablando de un catálogo de patrones paralelo donde los resultados de aplicar dichos patrones no son los que se preveían o que provoca mayor daño que beneficio por lo que no se aconseja su utilización. A este tipo de patrones se le denomina antipatrón [47] y constituyen un catálogo de experiencias que todos los expertos o no debieran tener en cuenta a la hora de afrontar un diseño de un espacio formativo.

Patrones de e-learning

En el ámbito del e-learning, son necesarios distintos tipos de profesionales para confeccionar un módulo de aprendizaje (pedagogos, programadores, expertos en diseños web, diseñadores institucionales, expertos en seguridad, accesibilidad, usabilidad, etc.) [67]. Esto hace que preparar un buen curso se convierta en una tarea titánica donde debieran confluír distintas sinergias. Contar con todos estos profesionales es prácticamente imposible, por ello, la utilización de “buenas prácticas” usadas por otros docentes afines, o por nosotros mismos en experiencias anteriores, se convierte en el procedimiento habitual para la elaboración de módulos de aprendizaje, es decir, aplicamos patrones de forma natural. A este problema debemos añadir que gran parte de los diseñadores y profesores autores de los cursos no han ejercido su actividad docente en entornos virtuales de aprendizaje, sino que proceden del ámbito presencial. En muchos casos, el proceso de adaptación del profesor consiste en un breve curso en el cual se definen las características generales de la enseñanza “online” y se recogen explícitamente las funciones que ha de desarrollar el profesor para la enseñanza de una asignatura o curso. No suelen abordarse aspectos que atañen de forma más particular al proceso de educación virtual: funciones propias del profesor o tutor “online”, diseño instructivo, temporalización y secuenciación, correcta aplicación de las herramientas de comunicación y evaluación de sistemas de gestión del aprendizaje, la evaluación del alumno, dinamización de foros, etc. Es posible encontrarse con numerosos casos de fracaso en la implementación de la enseñanza “online”. Precisamente, parte de este fracaso se debe a que suelen aplicarse los mismos métodos para la enseñanza online que para la enseñanza presencial.

El contexto del e-learning no ha sido ajeno a la utilización de patrones ni a los beneficios de éstos. Los patrones se pueden aplicar de manera casi natural porque los materiales didácticos y las herramientas de aprendizaje pueden mantener una estructura relativamente “estable”, que hace posible “registrar” una solución que se puede aplicar y volver a aplicar varias veces, es decir, que encaja perfectamente dentro de los objetivos de los patrones de diseño. Los patrones e-learning serán de utilidad para docentes que han de afrontar por primera vez la enseñanza en este entorno, pero también para docentes que deseen experimentar con nuevos modelos o simplemente que quieran cambiar su modelo para adaptarlo a nuevos retos e incluso a distintas herramientas o plataformas. Pero sobre todo deben ser útiles para dar solución a problemas específicos encontrados por el profesor o tutor en su actividad académica.

Los patrones e-learning pueden aplicarse para solucionar dificultades o problemas que surgen en el desarrollo de materiales de aprendizaje. De forma sencilla, los patrones permiten recopilar el conocimiento de expertos en áreas determinadas, buenas prácticas y el aprendizaje de las experiencias.

Pero de nada sirve almacenar toda esta experiencia si no elaboramos una forma de catalogar, documentar y poder consultar estas experiencias.

Los patrones e-learning no tienen por qué ser exclusivamente pedagógicos, también debe involucrar otro tipo de patrones como pueden ser web, de usabilidad, de seguridad, etc., puesto que al final se trata de elaborar un producto software que de soporte al modelo de aprendizaje elegido por el propio docente.

Existen numerosas iniciativas de aplicación de patrones para la mejora de los procesos de implementación de alternativas de aprendizaje virtual. Estas iniciativas se centran en distintos focos de atención para cubrir el ciclo de aprendizaje, desde la elaboración de materiales, utilización del software hasta los criterios pedagógicos empleados en distintas situaciones. Una de las principales áreas de aplicación es la producción de objetos de aprendizaje a partir de patrones (GLO Maker [35], WebQuest [36]) pero también se puede aplicar para mejorar los procesos organizativos en instituciones educativas. En este entorno, la Universidad Central de Queensland (Australia) ha desarrollado un marco de implementación y catalogación de patrones para mejorar las prácticas educativas en entornos virtuales [37].

En los últimos años están surgiendo distintos proyectos de catalogación de patrones e-learning en conferencias o publicaciones relacionadas con este enfoque. Entre todos ellos, cabe destacar: E-LEN [38], E-DILEMA[39], PCeL[40] y PPP[54].

El proyecto E-LEN tiene como objetivo desarrollar y difundir conocimiento sobre buenas prácticas para el diseño e implementación de entornos de aprendizaje soportados tecnológicamente. Propuesto para compartir experiencias entre centros asociados propone establecer la infraestructura y estructura organizativa de los centros, identificar buenas prácticas y elaborar y difundir recomendaciones para la creación de centros e-learning.

E-DILEMA recoge una colección de patrones pedagógicos de alto nivel y patrones de diseño de interfaces que proporcionan soporte a la hora de seleccionar y diseñar actividades de aprendizaje y cursos a través del diseño de páginas individuales. Enfocado principalmente a la modalidad de aprendizaje mixto o *blended learning*, en el que las actividades relacionadas con el desarrollo de cursos *on-line* son complementadas con clases presenciales. Propone distintas pautas desde la elaboración de materiales por parte del profesor, un repositorio para almacenar y difundir estos materiales y otros recursos y algunas propuestas para transformar experiencias de enseñanza tradicional a e-learning.

El proyecto PCeL parte de la concepción de que no existen dos personas que aprendan de la misma forma y a la misma velocidad. Desde esta idea, pretende combinar las ventajas del Aprendizaje Centrado en la Persona [53] con la utilización de las herramientas que nos ofrece el e-learning. Agrupa diferentes experiencias personales de actividades de aprendizaje en forma de patrones sobre tipos de cursos clásicos en términos de prácticas de e-learning, métodos para evaluar la actividad de los participantes, diferentes formas de recuperar información sobre cualquier aspecto del curso o de las actividades de aprendizaje que lo componen, o para fomentar la interacción e interactividad entre participantes, instructores, tutores y/o invitados.

En síntesis, las tres iniciativas pretenden crear un catálogo de buenas prácticas docentes aplicadas a e-learning desde diferentes puntos de partida. Para ello parten de buenas experiencias, las contrastan con otros docentes de disciplinas similares y ajenas, y finalmente se documenta esta experiencia de forma que todo el mundo pueda hacer uso de ella. Se basan en la utilización de un lenguaje estructurado que homogenice, por un lado el vocabulario empleado y por otro la información que de cada patrón se precisa. Las tres iniciativas pretenden crear patrones más o menos universales (participan varias universidades en cada una de ellas). Sin embargo, no consiguen ponerse de acuerdo en utilizar una estructura de descripción común. Eso complica la interacción entre los catálogos y muestra la importancia de utilizar una estructura de descripción de patrones común.

La idea de acercar los patrones pedagógicos al campo del e-learning no es nueva [76,77], pero en muchos casos, esta aproximación se limita a crear “artefactos” que convierten los patrones en procesos de enseñanza sin apenas intervención humana. Nosotros hemos pensado que la aparición de nuevas herramientas más sofisticadas no debe primar sobre la aplicación de criterios pedagógicos que han demostrado su efectividad durante mucho tiempo. Por ello abogamos por adecuar el uso de estas herramientas para la obtención de los fines claramente definidos en experiencias docentes y no al contrario. Desde este punto de vista, partimos de un catálogo de patrones pedagógicos ampliamente aceptados para trasladar estos a un ambiente de e-learning.

Patrones pedagógicos

Existen múltiples publicaciones con ejemplos de “buenas prácticas” docentes en distintos entornos educativos [94,95,97]. Todas ellas cuentan experiencias exitosas en diferentes escenarios e invitan al lector a su aplicación en su propia actividad docente. Cuando esto ocurre el resultado puede llegar a ser muy frustrante, pues lo que puede funcionar muy bien en un ambiente, puede producir pobres

resultados en otros. Como consecuencia, se produce un rechazo a utilizar estas “buenas prácticas” con el argumento de que nuestra área docente no se presta a la utilización de esas experiencias. Pero otras veces, el resultado de aplicar estas “buenas prácticas” es muy positivo a pesar de que el docente necesite adaptar la experiencia documentada a su propio entorno. Si este proceso de adaptación y los resultados de su aplicación pudieran ser incorporados al catálogo de “buenas prácticas” facilitaría la aplicabilidad y usabilidad de estos ejemplos para futuros interesados. De esta forma, estas buenas prácticas pueden llegar a convertirse en patrones según lo descrito anteriormente.

Un *patrón pedagógico* describe un problema que se presenta con frecuencia en el proceso de enseñanza-aprendizaje, con el fin de proponer a continuación una solución a ese problema que ha demostrado su efectividad en contextos similares, de modo que esa solución puede ser adoptada ante problemas semejantes [17]. Se trata de problemas relacionados con cualquier aspecto, momento o elemento del proceso: motivación de los alumnos, selección de contenidos, selección de materiales, secuencia de contenidos, selección de actividades, evaluación, procedimientos de evaluación, criterios de calificación, etc. Los patrones pedagógicos reflejan un catálogo de buenas prácticas que no necesariamente implican soporte tecnológico.

Algunos de los catálogos de patrones pedagógicos más referenciados son:

- **Educación en las aulas.** Anthony [30] describe catorce patrones, ordenados por grado de abstracción, de aplicación en las aulas para la enseñanza de materias técnicas. Los patrones más abstractos están dirigidos a aportar soluciones y los más concretos son de aplicación inmediata.
- **Grupos de estudio.** Kerievsky [31] afirma que los individuos deben ampliar sus conocimientos cada día para maximizar la productividad y la eficiencia de sus equipos. Todos los miembros de un equipo no aprenden las mismas cosas a la misma velocidad por lo que se hace necesario coordinar el aprendizaje en equipo. Desde esta óptica describe veintiún patrones organizados en cuatro secciones íntimamente ligadas destinados a propiciar un ambiente de colaboración dentro de cada grupo facilitando la tarea de aprendizaje.
- **Seminarios** [32]. Fricke y Völter [32] detallan 48 patrones a tener en cuenta para la preparación y desarrollo de seminarios. Utiliza una estructura basada en tres secciones para documentar cada patrón y los etiqueta en tres niveles según el grado de aceptación.

- **Tiempos de un curso.** Eckstein [33] propone cinco patrones pedagógicos de aplicación general relacionados con situaciones de enseñanza presencial como la manera de finalizar un curso, el modo de hacer menos dependientes del profesor a los alumnos, como favorecer la implicación de los alumnos en la búsqueda de sus propias soluciones, etc.
- **Pautas pedagógicas.** Bergin [34] indica catorce patrones de ámbito general detallando para cada uno de ellos una identificación (número, denominación, versión y breve descripción), problema, contexto, fundamentación, solución, discusión/consecuencias, recursos especiales, patrones relacionados, ejemplos, contraindicaciones y referencia.

Existen múltiples ejemplos de “buenas prácticas” en el ámbito educativo. El Proyecto de Patrones Pedagógicos (PPP) [54] pretende elaborar un catálogo de estrategias pedagógicas de manera que su lenguaje y estructuración no se aparte mucho de la forma en que los profesionales no afines al campo de la informática, documentan sus prácticas aulas o cursos virtuales. Dentro de este proyecto se enmarcan las propuestas de Bergin, Eckstein, Fricke y Völter mencionadas anteriormente.

A cada momento están surgiendo nuevos patrones pedagógicos que ofrecen soluciones probadas para problemas concretos dentro de distintos entornos. Sin embargo, no es objeto de este proyecto establecer una comparativa detallada de estos catálogos de patrones. Nuestra propuesta se centra en ofrecer pautas para adaptar un catálogo de patrones ampliamente aceptado dentro del entorno donde se aplique y convertir estos patrones en propuestas para adaptar estos patrones a un ambiente e-learning. Desde este punto de vista, hemos seleccionado las pautas pedagógicas de Bergin por varios motivos:

- Constituye un número reducido (catorce) de pautas con nombres muy sugerentes que permiten identificar claramente la propuesta de su aplicación.
- Ofrece un lenguaje suficientemente abstracto como para poder ser entendido por docentes de diferentes disciplinas.
- Son patrones perfectamente documentados siguiendo una estructura o lenguaje común para todos ellos.
- Todos los patrones son aplicables a entornos e-learning.
- Existe una interrelación entre patrones mostrada en la Figura 2.

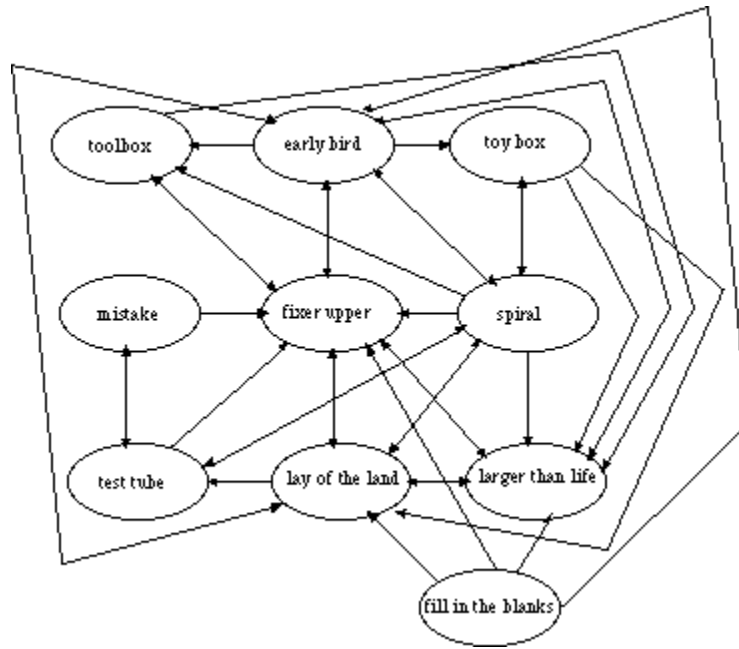


Figura 2. Relación entre patrones de Bergin.

Los catorce patrones de Bergin se exponen en el capítulo 4, y se incluye la documentación estructurada de cada uno de ellos. Es interesante destacar que Bergin intenta mantener un lenguaje cercano al profesor no experto en el uso de TICs, a pesar de que todos ellos tienen una aplicabilidad en entornos e-learning.

3. Funciones de un LMS genérico

La sección anterior se centraba en patrones, y en particular en los patrones pedagógicos propuestos por Bergin. El objetivo de este trabajo es proporcionar soporte a través de un LMS a dichos patrones de Bergin. Esta sección va a identificar las funciones de un LMS genérico, con el fin de caracterizar las herramientas de un LMS que pueden dar soporte a los patrones de Bergin. De esta forma se ganará en nivel de abstracción, al desvincular el soporte a patrones de LMSs concretos.

Desde la perspectiva del profesor y del creador de contenidos los LMS o LCMS contienen diferentes tipos de herramientas de autor. Por ejemplo, puesto que los LMS se basan en entornos Web, gran parte de los documentos utilizan el lenguaje HTML para su presentación, por ello, suelen ofrecer editores WYSIWYG (*What You See Is What You Get*) para facilitar la elaboración de estos contenidos. De cualquier forma, las herramientas de autor ofrecidas en cada LMS son una parte importante del proceso de e-learning. Basándonos en la división de herramientas proporcionada por Boneu [41], podemos agrupar las herramientas de los LMS de la siguiente manera:

I. Herramientas orientadas al aprendizaje

- I.1. **Foros.** Permiten una comunicación asíncrona entre usuarios. Normalmente se organizan en torno a temas concretos (hilos). Los usuarios pueden contestar (*postear*) a estos temas u otras aportaciones realizadas creándose una línea de colaboración entre los usuarios. Puesto que la comunicación es asíncrona, la mayoría de los LMS prevén un medio de notificación al usuario que facilite el reconocimiento de estas aportaciones (p.e.: RSS, SMS, e-mail, avisos). Las aportaciones a los foros pueden incluir documentos adjuntos, enlaces externos e incluso formato HTML. Otras opciones son: editores HTML, editores de fórmulas matemáticas y correctores ortográficos.
- I.2. **Buscador de foros.** Una herramienta buscador en los foros se convierte en fundamental cuando el número de aportaciones es alto. La herramienta puede aceptar distintos parámetros de búsqueda de forma que dirijan al usuario en su búsqueda seleccionando los post que más se adecuen a la preferencia del usuario. La mayor parte de los buscadores

permite realizar distintos tipos de búsqueda: por autor, por fechas, por hilos, por ubicación. Además permiten utilizar un lenguaje de búsqueda que permita usar operadores lógicos en las consultas.

I.3. E-portafolio. Un portafolio se basa en una carpeta donde el alumno deposita materiales que pueden ser evaluados por el docente. Pero un portafolio es mucho más. Se trata de que el docente no solo evalúe los trabajos, sino la forma en que estos se han elaborado. Con ello, se consigue un doble objetivo: que los docentes conozcan el resultado de su método pedagógico sobre el alumno y que el alumno reflexione sobre su propio aprendizaje y se sienta dueño de su progreso [42]. Los portafolios son ampliamente utilizados en otras disciplinas, pero es en ámbito educativo donde cobran más fuerza. No se trata solo de que el alumno exponga sus trabajos, sino que documente como ha desarrollado estos trabajos de forma que el docente evalúe el esfuerzo y la trayectoria para la conclusión de estos trabajos.

I.4. Intercambio de archivos. El intercambio de archivos permite a profesores y alumnos compartir documentos propios o ajenos dentro del LMS. Esta funcionalidad también puede realizarse desde otras herramientas del LMS: foros, correos, tareas, blogs, e-portafolio). La forma de incorporar archivos a la plataforma puede ser mediante funciones para subir archivos (Drop-box)⁴ o mediante WebDAV⁵, que permite gestionar el contenido de un curso como una carpeta local del ordenador del usuario. Este intercambio de archivos puede realizarse desde un repositorio común o como añadido a alguna otra funcionalidad, pero en cualquier caso, debe permitir a emisor y destinatario recuperar el archivo en cualquier momento. La herramienta de intercambio debe permitir cualquier formato: audio, video, texto, animaciones, etc. Habitualmente será el propio destinatario quien deba de disponer de la aplicación o plug-in destinada a visualizar cada tipo de archivo y si no fuera así se le recomendará donde descargar esa aplicación (wav, avi, flash, etc.). En otros casos, el formato será propietario y necesitará de un software, normalmente comercial, específico para su tratamiento (MSoffice, Autocad, etc.). Los tipos de documento más

⁴ <http://es.wikipedia.org/wiki/Dropbox>

⁵ <http://www.webdav.org/>

elementales pueden ser visualizados dentro del propio navegador web (imágenes, HTML, pdf, txt).

- I.5. Herramientas de comunicación síncrona.** Dentro de este apartado se incluyen todas las herramientas que permite una comunicación interactiva entre usuarios (entre dos o entre varios) en el mismo momento. Se trataría de chats, videochats, Mensajería instantánea, VoIP, y otras tantas. La finalidad es conseguir la inmediatez en las comunicaciones que puede ser utilizada con distintos fines: tutoría virtual, una conferencia participativa, debate en línea, etc. Este tipo de herramientas están avanzando enormemente en posibilidades y complejidad.
- I.6. Herramientas de comunicación asíncrona.** Como complemento a las anteriores, existen muchas herramientas que no requieren de esa comunicación instantánea pero que permite mantener la comunicación entre los participantes. A este segmento pertenecería el correo y la mensajería. Los LMS permiten intercambiar mensajes entre alumnos de un mismo curso o de toda la comunidad. Estos mensajes, pueden contener formato HTML, enlaces externos y ficheros adjuntos. Pueden estar dirigidos a un individuo concreto o a un grupo de individuos e incluso a individuos con un rol concreto. Este correo interno puede o no estar conectado con el mail externo del usuario.
- I.7. Servicios de presentación multimedia.** Son servicios de comunicación más avanzados que los anteriores (videoconferencia participativa, pizarra electrónica). Estos servicios se refieren al uso de videoconferencia entre el sistema y el usuario, o a la comunicación entre dos usuarios cualesquiera. Una pizarra electrónica puede ser utilizada por el profesor con sus estudiantes en una clase virtual; este es un servicio de comunicación síncrona entre profesores y estudiantes, tal y como puede ser también la compartición de aplicaciones, que residen en uno de los equipos de los usuarios, con el resto de usuarios. Habitualmente, este tipo de herramientas permite al administrador cambiar el rol de los participantes para asignarles distintas funciones y simular el comportamiento en un aula presencial.
- I.8. Diario (blog) / Notas en línea.** El Blog es una herramienta que puede ser utilizada por estudiantes y profesores para efectuar anotaciones en un diario. La mayor facilidad de los blogs es que cualquier persona puede crear una contribución con elementos multimedia sin necesidad de conocer HTML ni ningún otro lenguaje específico, por tanto, los blogs suelen ir acompañados de editores WYSIWYG (*What You See Is What You Get*) para facilitar la

elaboración de estos contenidos. Este es aplicable en su aspecto más educativo a través de los edublogs⁶ que permiten clasificar los blogs en tres grupos distintos dependiendo de la audiencia:

- Blogs de asignaturas, en las que el profesor va publicando noticias sobre la misma, solicitando comentarios a los alumnos, propuestas de actividades, etc. En este tipo de blogs solo el profesor puede proponer hilos de conversación.
- Blogs individuales del alumno. En este tipo de blogs el alumno, de forma voluntaria o dirigida por el profesor/es debe ir realizando entradas periódicas para la evaluación o seguimiento. Los profesores u otros alumnos pueden realizar aportaciones para dirigir o colaborar con actividad que el alumno esté desarrollando.
- Blogs grupales. A medio camino, el profesor o el propio alumno, puede inscribir/participar en grupos dentro de la propia asignatura o de ámbito social como asociaciones, clubs, etc. De esta forma se potencia el trabajo en grupo de los alumnos y permite conocer al profesor su participación en estos grupos.

I.9. Wikis. Son herramientas que facilitan la elaboración de documentos en línea de forma colaborativa. Gracias a los wikis, el conocimiento ya no se basa solo en las fuentes clásicas, sino que es posible encontrar una diversidad amplia de matices que lo están haciendo más subjetivo. Un ejemplo bien claro de esto es la Wikipedia⁷. Al tratarse de un trabajo colaborativo, los wikis deben contemplar la figura del moderador y controlar las aportaciones realizadas por cada usuario. Un glosario es un ejemplo claro de utilización de wikis.

I.10. Encuestas. En la formación en línea se hace más necesario conocer la opinión de alumnos para adecuar el ritmo de trabajo o para proponer trabajos, prácticas, salidas. Las encuestas pueden ser anónimas o con identificación y normalmente suelen estar limitadas en el tiempo. Algunos LMS proponen realizar encuestas de forma síncrona de manera que los alumnos contestan en un intervalo pequeño de tiempo y el profesor recibe estos resultados en el momento. En este caso, el profesor puede definir el tamaño de la población, de

⁶ <http://edublogs.org/>

⁷ <http://es.wikipedia.org/wiki/Wikipedia>

forma que no sea necesario recabar la opinión de todos los estudiantes. Es habitual que los participantes puedan conocer los resultados globales de la encuesta durante o al final de la realización.

II. Herramientas orientadas a la productividad.

- II.1. Anotaciones personales o favoritos.** Las anotaciones o *bookmarks* se utilizan para poder volver fácilmente a una página web visitada. Desde este punto de vista, las anotaciones pueden ser privadas o públicas para facilitar el acceso de los alumnos a ciertas páginas web. De igual modo, pueden ser asociadas a un curso o individuales del propio usuario.
- II.2. Calendario y revisión del progreso.** El calendario permite al alumno planificarse con respecto a las actividades del curso. En esta herramienta se registran el calendario de exámenes y tareas, las actividades docentes: charlas, visitas, conferencias, *webminar* (seminario web), etc. Además, el alumno puede incluir sus propias citas de forma manual o por invitación. Normalmente, los LMS permiten al alumno disponer de un calendario global donde se resuman todas las anotaciones de todas las asignaturas. La herramienta de revisión del progreso permite al alumno conocer el grado de cumplimiento de la planificación de la asignatura.
- II.3. Buscador de cursos.** Este tipo de herramientas permite a los alumnos la selección y localización de los cursos indicando un patrón de búsqueda. Podrá ser de gran utilidad no solo para los procesos de selección de asignaturas, también para localizar asignaturas con acceso “abierto” que nos sirvan como complemento a otras asignaturas.
- II.4. Mecanismos de sincronización y trabajo fuera de línea.** Los estudiantes tiene posibilidad de trabajar desconectados de la plataforma, para ello, el estudiante debe poder descargarse el curso, o parte de él, en su ordenador, y trabajar localmente en el curso, de modo que la próxima vez que acceda a la plataforma, se sincronizará o se actualizará el punto en el que se encontraba el estudiante en su estudio, la última vez que se desconectó. Este tipo de herramientas permite el progreso del alumno, consumiendo materiales, realizando ejercicios, desarrollando nuevos materiales sin necesidad de estar conectado en todo momento.

II.5. Control de publicación, páginas caducadas y enlaces rotos. Estas herramientas permiten publicar materiales (páginas, exámenes, material multimedia, etc.) al llegar a unos objetivos determinados como pueden ser fechas, calificaciones de exámenes, cumplimentación de formularios. De igual modo pueden establecerse criterios para la desaparición de estos materiales. Otra herramienta es el control de enlaces rotos, normalmente motivados por la eliminación / reubicación de los materiales dentro del mismo curso.

II.6. Noticias del lugar. Estas herramientas permiten mantener informado de las últimas novedades al usuario de la plataforma, por tanto, son consideradas de ámbito global y normalmente no suelen ser de carácter educativo.

II.7. Envío de actualización de páginas, mensajes a foros y envío automático. Cada usuario puede participar en distintos cursos lo que hace muy difícil mantenerse informado de las modificaciones realizadas en cada curso. Los LMS suelen contener herramientas de notificación de novedades dentro de cada curso. Los avisos que se generan varían enormemente de unas plataformas a otras, pero en general se refiere a:

- Aportaciones a foros.
- Envío de trabajos.
- Mensajes.
- Citas del calendario
- Disponibilidad de nuevos materiales
- Exámenes
- Calificaciones

II.8. Soporte a la sindicación de contenidos. Estas herramientas permiten incorporar a la plataforma contenidos de forma sindicada, que son ofrecidos desde el exterior o desde la propia plataforma, incluso crear contenidos que puedan sindicarse desde otras plataformas. Ejemplo de estas herramientas son los RSS (*Really Simple Syndication*), los *Podcasts* o *News*. Gracias a los agregadores o lectores de *feeds*, se pueden obtener resúmenes de todos los sitios que se desee. Las RSS se refieren normalmente a contenidos textuales, mientras que los *podcast* se refieren a contenidos multimedia (imagen y/o sonido), mediante sindicación. La sindicación en ambos casos se realiza utilizando archivos en formato

XML. Para recibir las novedades, se precisa un lector de RSS o PodCast que compruebe si existe algún feed nuevo que sea de nuestro interés según un filtro preestablecido. De esta forma evitamos saturar el correo con notificaciones de novedades. Los *news* o *newsgroups* son utilizados para enviar/recibir artículos de distinta temática. Técnicamente difieren de los Foros de discusión mostrados anteriormente, pues las aportaciones residen en distintos servidores y están jerarquizadas por temáticas; sin embargo, funcionalmente, tienen el mismo comportamiento pues el usuario puede leer artículos de otros usuarios y contestar a todos los usuarios. Para leer/contestar estas aportaciones se precisa de un lector de news (Newsreader) [43]. Un Podcast es una grabación de audio y/o video disponible en línea. El término deriva de la combinación de las palabras iPod (Reproductor mp3 de Apple) y Broadcast, aunque para reproducir o generar una de estas grabaciones no es necesario un iPod o cualquier otro dispositivo de Apple. El contenido de los Podcast puede ser de todo tipo, aunque los más comunes son los de audio, también pueden utilizarse presentaciones (tipo Powerpoint) con sonido intercalado e incluso podcast de video.

III. Herramientas para la interacción/participación de los estudiantes.

III.1. Grupos de trabajo. Los grupos de trabajo ofrecen la posibilidad de organizar una clase en grupos, de forma que proporciona un espacio y una comunicación entre todos los miembros del grupo. El profesor puede asignar una tarea a cada uno de estos grupos de trabajo y participar o no dentro de cada uno de ellos. Los grupos de trabajo pueden tener asociadas ciertas herramientas para su uso exclusivo (foros, correos, intercambio de archivos, etc.). Los grupos pueden ser privados o abiertos donde los usuarios pueden participar dentro de estos grupos.

III.2. Autovaloraciones. Con este tipo de herramientas, el estudiante puede practicar y valorar por sí mismo su propia capacidad, así como la calidad del trabajo realizado. El estudiante puede medir su progreso adquiriendo más responsabilidad sobre él e incrementar el nivel de motivación para llegar al nivel de competencia deseable [44]. Estas autoevaluaciones pueden consistir en tests en línea que proporcione a alumno retroalimentación sobre las preguntas fallidas, pero también pueden usarse simuladores o animaciones flash que presenten un reto al alumno. Es de especial interés durante el primer año de universidad,

donde se ayuda más a los estudiantes a que se adapten a la vida universitaria. Normalmente estas pruebas no son contabilizadas por el profesor.

III.3. Rincón del estudiante. También denominado rincón social. Estas herramientas permiten al estudiante participar en grupos y clubs de su interés fuera de sus asignaturas, manteniendo un trabajo colaborativo en áreas de su interés.

III.4. Perfil del estudiante. Son espacios donde el estudiante puede mostrar su trabajo, anunciarse, mostrar su fotografía, preferencias, temas de interés, formas de contacto en redes sociales e información personal. El estudiante puede limitar la información que de él se ofrece para el resto de estudiantes o para los alumnos con los que comparte asignaturas. Esta información puede ser ofrecida al resto de estudiantes y complementada por el propio profesor de forma que pueda confeccionarse un listado de participantes u orla del curso.

IV. Herramientas de soporte.

IV.1. Autenticación de usuarios. La autenticación es el proceso a través del cual se proporciona acceso a un usuario a un curso, con identificador del usuario y contraseña. La forma más elemental es mantener los registros de los usuarios dentro del propio LMS. Un segundo método consiste en sincronizar periódicamente los registros de los usuarios con la aplicación de matrícula utilizada por la institución. Un tercer método consiste en delegar en un tercero la validación del usuario y contraseña estableciendo relaciones de confianza (LDAP - *Lightweight Directory Access Protocol*).

IV.2. Asignación de privilegios en función del rol del usuario. Un rol es un conjunto de permisos que puede asignarse a un usuario; por su parte, un permiso es un parámetro que especifica si su poseedor dispone de acceso a una determinada función del sistema o a una parte de la interfaz de usuario del sistema. Normalmente, los roles se definen de modo que incluyan permisos que guarden cierta relación y suelen corresponderse con algún rol de la vida real. Se pueden asignar varios roles a un mismo usuario. De este modo, se garantiza que el usuario disponga del conjunto de permisos definidos por tales roles. El LMS efectúa la asignación de tales permisos al usuario cuando éste se conecte al sistema. La asignación sólo permanecerá en vigor mientras dure la sesión. Esto significa que, si se modifica un rol mientras está conectado el usuario al que se le ha asignado, la modificación no surtirá

efecto hasta que ese usuario se desconecte y vuelva a conectarse. Los LMS distinguen entre dos tipos de roles: aplicados y automáticos dependiendo de si ha sido asignado por el propio usuario o por el LMS. En general, en los LMS participan profesores y alumnos cuyos papeles pueden ser intercambiados para facilitar el proceso de aprendizaje. La gestión de qué herramientas puede emplear uno y, otro y de qué manera acceder a ellas se resuelve empleando roles que puedan ser intercambiables. Habitualmente los roles son modificables y adaptables a la forma de trabajar de cada docente o institución. (Profesor, creador del curso, ayudante, alumno, etc.).

IV.3. Registro de estudiantes. En general, los LMS están integrados con aplicaciones de matriculación externas, pero también es habitual disponer de herramientas para facilitar la inscripción abierta en asignaturas donde el usuario pueda inscribirse voluntariamente (con o sin contraseñas) durante un periodo de tiempo e incluso que sea el profesor quien pueda agregar o inhabilitar el acceso a los alumnos a su asignatura.

IV.4. Auditoría. Las herramientas de auditoría permiten consultar todas las acciones realizadas por los participantes, así como obtener estadísticas sobre su utilización. Estas herramientas pueden ser utilizadas por administradores del LMS para verificar el uso que se hace del sistema, pero también puede ser utilizadas por el profesor para conocer los contenidos más usados y si el alumno tiene problemas o no para llegar hasta ciertos contenidos. También existe la posibilidad de extraer estos informes en distintos formatos para su tratamiento estadístico externo.

V. Herramientas destinadas a la publicación de cursos y contenidos.

V.1. Test y resultados automatizados. Los profesores pueden administrar los test que propongan ofreciendo condiciones de uso, aleatoriedad de las preguntas y ofrecer retroalimentación, comentarios o enlaces relacionados dependiendo de las respuestas o resultados obtenidos por el alumno.

V.2. Administración del curso. Estas herramientas permiten al profesor modificar presentaciones del curso y conocer la progresión de los alumnos dentro de la clase. También permite a los alumnos comprobar sus progresos, con los trabajos, test, pruebas, etc.

- V.3. Apoyo al creador de cursos.** Se trata de ofrecer ayudas y apoyo a los creadores de cursos en la administración de estos. Estas ayudas pueden venir a través de foros, chats, FAQs, ayudas en línea, por teléfono, correo electrónico, etc.
- V.4. Herramientas de calificación en línea.** Estas herramientas permiten al profesor conocer el seguimiento y trabajo del estudiante dentro del curso.
- V.5. Seguimiento del estudiante.** Permite conocer la participación del estudiante dentro de cada curso (no solo conocer que herramientas del LMS ha utilizado, que lo contempla las herramientas de auditoría, sino como las emplea). Aportaciones a los foros, wikis, blogs, trabajos presentados, grupos en los que participa, etc.

VI. Herramientas para el diseño de planes de estudio

- VI.1. Conformidad con la accesibilidad.** Esto implica cumplir con los estándares y/o recomendaciones que permitan que las personas con alguna discapacidad puedan acceder a la información en línea.
- VI.2. Reutilización y compartición de contenidos.** El sistema debe permitir compartir un sistema de archivos y repositorios de contenidos abiertos. Los contenidos creados por un curso deben poder compartirse con otros cursos incluso de otros profesores.
- VI.3. Plantillas de curso.** Esta herramienta permitirá a los diseñadores de cursos reutilizar plantillas de cursos creados por otros o por él mismo.
- VI.4. Administración del currículo.** Permite proporcionar un currículo personalizado a los estudiantes, basado en los prerrequisitos del programa educativo o actividades, en trabajos previos o resultados de test
- VI.5. Personalización del entorno (look & feel).** El profesor debe poder variar la apariencia gráfica y como se ven los cursos, de esta forma, el profesor puede combinar la apariencia de su curso con la imagen de la institución en los mismos.
- VI.6. Herramientas para el diseño de la educación.** Herramienta de ayuda a los creadores de cursos para poder crear secuencias de aprendizaje, plantillas o asistentes.
- VI.7. Conformidad con el diseño de la educación.** Conformidad con los estándares (IMS, AICC y ADL, básicamente) para la compartición de materiales de aprendizaje con otras plataformas de e-learning [93].

VII. Sistemas para la gestión del conocimiento en el ámbito educativo

Reflejamos en esta categoría sistemas que sin ser propiamente herramientas de los LMS, aparecen cada vez más ligados a las plataformas educativas.

VII.1. Sistemas integrales de conocimiento. Son sistemas utilizados para representar la estructura organizativa del conocimiento mediante trabajo colaborativo así como organización interna de la memoria del conocimiento.

VII.2. Sistemas mediadores de información. La finalidad de estos sistemas es el de actuar como interprete entre las necesidades de información del usuario y los contenidos disponibles en distintas fuentes, de forma que el usuario no perciba la heterogeneidad de estas fuentes. Para la descripción e indexación de las fuentes de conocimiento y su contenido, se utilizan lenguajes que permitan relacionar la información como el caso del estándar RDF (*Resource Description Framework*). La utilización de este tipo de lenguajes pretende convertir la Web de archivos en una Web donde los datos recuperen el interés y no sea necesario un programa específico para extraer información de estos datos. A esto se le denomina Web Semántica. [45]

VII.3. Librerías digitales o repositorios. Son sistemas que permiten el almacenamiento digital de recursos educativos en general, desde simples textos a cursos estructurados con algún estándar de los indicados en el punto VI.7. Estos repositorios surgieron como necesidad para compartir materiales educativos de calidad entre profesionales de distintos centros. Algunos de los proyectos más conocidos son Merlot, Internet Archive o Agrega en España. Paralelamente al archivo de estos materiales se precisa herramientas de búsqueda, y en consecuencia, de catalogación de recursos.

VII.4. Sistemas basados en ontologías. En este tipo de sistemas se estructura el conocimiento en base a reglas conocidas, esto permite que ciertos elementos de conocimiento se solapen en más de un sitio, pero facilita la búsqueda de información. Técnicamente consiste en ubicar estos objetos en un repositorio siguiendo reglas de ordenación definidas de antemano.

VII.5. Sistemas basados en folksonomías. Consiste en el proceso contrario, las reglas no son conocidas a priori y es el usuario quien etiqueta y por tanto, clasifica los elementos de información según su criterio. Pudiera parecer un sistema algo caótico pero la experiencia

demuestra que son sistemas son ampliamente utilizados y permiten libertad y capacidad de adaptación que no permiten las ontologías.

4. Propuesta de funciones para patrones

En el capítulo anterior hemos analizado los tipos de herramientas que compondrían un LMS genérico, si bien, tal como se indicaba, esta clasificación de las herramientas no presenta una línea divisoria clara para la categorización de ciertas utilidades que ofrecen muchos de los LMS de mayor éxito. De hecho, algunas herramientas podrían pertenecer a varias categorías distintas. Sin embargo, se hace necesaria realizar esta categorización para que el usuario pueda decidir qué tipo de herramientas pueden ser útiles para sus necesidades formativas y educativas con independencia del LMS que ofrece su institución.

Por otro lado, la presencia de patrones pedagógicos, necesarios para la obtención de resultados formativos positivos, no garantiza su correcto uso en un entorno e-learning. Por tanto, cobra especial importancia el guiar al docente en la implantación de dichos patrones pedagógicos utilizando las herramientas de un LMS. En la medida que las herramientas pertenezcan al LMS genérico considerado en el capítulo anterior, se obtendrá una mayor generalidad e independencia de la plataforma educativa subyacente.

En el capítulo 2 se han mencionado distintos catálogos de patrones pedagógicos ampliamente utilizados en diferentes entornos, y se ha justificado la selección de los patrones de Bergin para nuestra experimentación. La adopción de un catálogo de patrones u otro deberá ser consensuado, e incluso confeccionado, por la propia institución de forma que se adapte a sus necesidades formativas. El siguiente paso consiste en definir las herramientas básicas que deberían emplearse para la aplicación de cada patrón Bergin con instrucciones básicas de como emplearlas. Es importante recalcar que se trata de herramientas básicas, pues el docente podría utilizar herramientas adicionales para adaptar el patrón a sus propios objetivos.

En este proyecto hemos definido una tabla cartesiana que muestra, por un lado los patrones seleccionados y, por otro, las herramientas básicas para la aplicación de cada patrón y hemos creado una pequeña guía para la aplicación de estas herramientas. Esta sería una primera aproximación a la aplicación de patrones que deberá ser consensuada y cumplimentada en la medida en que los patrones sean utilizados por distintos docentes. Cabe destacar que en esta tabla no hemos propuesto ninguna herramienta orientada a la productividad, ya que están pensadas para que el alumno las utilice discrecionalmente, sin intervención del profesor.

Bergin propone 14 patrones ampliamente documentados, pero no indica cómo aplicar estos patrones en un ambiente de e-learning, por tanto, si el lector no es experimentado en el uso de las herramientas de cualquier plataforma LMS, puede no encontrar la forma de aplicar estos patrones. En las siguientes fichas, se describen los catorce patrones con la información detallada de cada uno de ellos. Sobre estas fichas realizamos la propuesta de selección de las herramientas de un LMS genérico necesarias para la aplicación de los patrones, junto con una guía de cómo utilizar estas herramientas. Es importante indicar que, para cada patrón, se muestran un número reducido de herramientas que se consideran más importantes para la aplicación, pudiendo el diseñador interesado incluir nuevas herramientas que se adapten mejor a su planteamiento docente. El orden de propuesta de cada herramienta indica el orden de prioridad en la misma. También conviene mencionar que Bergin realiza su actividad docente como profesor de informática, por lo que la mayoría de los ejemplos y referencias están ligados a esta disciplina, si bien, el problema y solución pueden trasladarse a cualquier otra disciplina. Por último cabe destacar que en las tablas siguientes pueden aparecer referencias a patrones y personas no considerados en este trabajo, ya que los patrones analizados en este trabajo son un subconjunto distinguido seleccionado por el propio Bergin de un catálogo mayor. Dichos patrones y personas pueden encontrarse en [34].

Nombre del Patrón:	EARLY BIRD	#1
Descripción	Organizar el curso para que los temas más importantes se enseñen en primer lugar. Enseñar el material más importante, las "grandes ideas", al principio (y frecuentemente). Cuando esto sea imposible, enseñar el material más importante lo antes posible	
Problema	Un curso típico tiene muchos temas importantes. Muchas veces están relacionados entre sí. Es difícil decidir cómo ordenar los temas para que los estudiantes pueda apreciar las "grandes ideas" en el curso. Si se demoran los temas importantes hasta el final del curso, pasando mucho tiempo en preliminares, los estudiantes pueden tener una idea equivocada de la importancia relativa de estos temas. También se pueden fortalecer las grandes ideas por medio de ejercicios y discusiones.	
Audiencia/Contexto	Es posible aplicarlo en casi todos los ámbitos.	
Fuerzas	<p>Los estudiantes necesitan ver hacia dónde se dirigen. Ellos necesitan ver que los detalles presentados en las primeras etapas se relacionan con las ideas importantes.</p> <p>Los estudiantes necesitan saber cuáles son las ideas grandes de cada curso. Tienen que ser capaces de separar los conceptos clave de los detalles que los apoyan.</p> <p>Los estudiantes suelen recordar mejor lo que aprenden en primer lugar. Esto puede ser tanto positivo como negativo. Las ideas importantes (grandes) deben ser introducidas desde el primer momento, incluso si no pueden abarcarlas por completo.</p>	
Solución	<p>En primer lugar identificar las ideas más importantes en el curso. Rastrear el curso en busca de estas ideas. Estas ideas se convierten en el principio fundamental de la organización del curso. Introducir estas grandes ideas, y sobre todo sus relaciones al inicio del curso y volver sobre ellas repetidamente a lo largo del curso.</p> <p>Ordenamos los temas de clase en orden de importancia y encontrar maneras de enseñar las ideas más importantes al principio.</p> <p>Si esto no es posible por diversas razones, buscar la forma de introducir estas ideas importantes lo antes posible.</p>	
Discusión / Consecuencias / Implementación	<p>Los asuntos más importantes en un curso o programa de estudios reciben más atención por parte del instructor y los estudiantes, si se tiene la oportunidad de volver sobre ellos una y otra vez. Los estudiantes pueden ser más conscientes de lo que es de suma importancia.</p> <p>La aplicación es difícil. Con frecuencia sólo podemos introducir desde el principio aspectos simples de una idea importante. Algunas veces es suficiente mostrar los términos más importantes y las ideas generales. Cuando las "grandes" ideas son consideradas demasiado complejas, es difícil mostrarlas al principio. Se precisa de una reflexión y preparación del diseño curricular. A veces, un concepto realmente grande, pero difícil, puede ser introducido de forma incompleta, entonces, como cualquier otro material afín, la relación debe ser cuidadosamente explorada.</p> <p>Los profesores deben ser capaces de analizar cuáles son las consecuencias de desarrollar el material en un orden particular. A menudo es útil disponer de un foro en el que poder discutir y refinar las ideas. Con frecuencia es necesario desarrollar tus propios materiales, lo que requiere tiempo y esfuerzo.</p>	
Requisitos	Se necesita reflexión y disponer de tiempo. También es importante la participación en grupos de discusión con otros educadores que compartan ideas similares acerca de los conceptos más importantes.	
Patrones relacionados	Es posible que sea necesario el patrón <i>spiral</i> para introducir algunos conceptos previos.	

Nombre del Patrón:	EARLY BIRD	#1
<p>Para mostrar a los estudiantes una gran idea en acción, se puede hacer uso de algún ejemplo de <i>Lay of the Land</i>. Si existen muchas ideas importantes que puede hacer uso de <i>Larger than life</i>. El patrón <i>Fixer upper</i> puede ser una buena manera de empezar. Se debe hacer hincapié en la gran idea. Si la idea es compleja, utilice el patrón <i>Toy Box</i> para introducirla. Ideas relacionadas entre sí a menudo puede conducir a los componentes de una caja de herramientas. Nota: Este patrón es recursivo, puesto que los patrones son en sí mismos grandes ideas</p>		
Ejemplos		
<p>Enseñar objetos al principio. Enseñar diseño desde el comienzo. Tratar la concurrencia al principio en sistemas operativos. Mostrar las necesidades de usuario en bases de datos. Enseñar a la recursividad antes que bucles. Por supuesto, estas son opiniones de Bergin sobre lo que es más importante. Cada docente puede discrepar, y por ello, debe encontrar las “primeras ideas” en su curso y ponerlas en práctica. El libro de Karel the Robot, de Richard Pattis fue diseñado con la idea de este patrón como una forma de enseñar programación basada en procedimientos. Su sucesor, Karel ++, continuó esta idea ampliándolo al uso de objetos.</p>		
Contraindicaciones		
<p>Será un error utilizar este patrón con materiales que requieren unos conceptos previos para la presentación de las ideas importantes. Con mayor motivo si las relación entre los conceptos previos y las ideas importantes no son claras o difíciles de entender. De nuevo, los patrones <i>Tool Box</i> y <i>Lay of the Land</i>, pueden servirnos para introducir estos temas.</p>		
Referencias		
<p>Karel the Robot, Richard Pattis, Wiley, 1981 Karel ++, Joseph Bergin, Mark Stehlik, James Roberts, Richard Pattis, Wiley, 1997.</p>		
Herramientas LMS propuestas		
<ul style="list-style-type: none"> • I.4 - Intercambio de archivos • V.6 - Herramientas para el diseño de la educación (secuencias de aprendizaje) • III.2 - Autovaloraciones • I.10 - Encuestas 		
Propuesta utilización		
<p>Los documentos que se ofrecen deben incluir un resumen o notas al margen para facilitar su comprensión (el autor no conoce ninguna herramienta de los LMS que permita esta opción). Es importante organizar el temario por secciones/temas (secuencias de aprendizaje) y disponer de autoexámenes/exámenes en cada sección o lección para valorar el aprovechamiento de los conocimientos adquiridos entre ellos. Es preciso definir un guión del curso de forma que el alumno pueda seguirlo sin perderse. Al final, se propondrán encuestas para conocer que partes de la asignatura han planteado mayor problema de comprensión. Otras herramientas pueden ser de utilidad como foros para comprobar la comprensión del alumno, auditorías para conocer el comportamiento de los alumnos ante ciertos conceptos, etc.</p>		

Nombre del Patrón:	SPIRAL	#2
Descripción	<p>Los temas en un curso se dividen en fragmentos y los fragmentos deben ser introducidos en un orden que permita la solución de problemas por los estudiantes. Muchos fragmentos sirven para introducir un tema, pero no entran en detalle. Al principio sólo se profundiza lo suficiente para obtener el conocimiento básico para resolver los problemas. En ciclos posteriores se ofrecen fragmentos que ayuden a profundizar sobre el tema.</p>	
Problema	<p>En un curso, con frecuencia los temas están relacionados entre sí. Normalmente, se precisan conocer varios temas para que los estudiantes tengan las herramientas necesarias para resolver problemas. Sin un orden lógico en la introducción de temas, los estudiantes pueden desorientarse y llegar a aburrirse. Los estudiantes deben, desde el principio del curso, estar capacitados para resolver problemas relevantes.</p>	
Audiencia/Contexto	<p>Este patrón se aplica a cualquier curso en el que hay un gran número de conceptos que se deben dominarse en conjunto.</p> <p>Este patrón se puede utilizar en varios tipos de curso, especialmente en los primeros cursos del plan de estudios. Se puede usar en los cursos de programación, análisis y diseño, y cursos especiales en tecnología de objetos. Una variación de este modelo se utiliza a menudo en el curso de construcción de compiladores.</p>	
Fuerzas	<p>Muchos campos sólo pueden ser dominados por el dominio por separado de un gran número de diferentes técnicas que han de interactuar.</p> <p>Grandes temas como la programación y el diseño requieren muchas partes y muchos detalles difíciles de dominar. El desarrollo de estos de una manera secuencial deja a los estudiantes sin ejercicios interesantes, ya que no han visto con suficiente amplitud el tema para hacer cosas interesantes.</p> <p>A los estudiantes les gusta construir cosas y les gusta ver cómo encajan las piezas. Se aburren fácilmente si la instrucción es repetitiva y si el instructor pasa demasiado tiempo con un tema o un conjunto de temas relacionados entre sí. Los estudiantes también pueden llegar a aburrirse si los ejercicios propuestos son artificiales para ilustrar detalles pasados.</p> <p>Los cursos no tienen que ser organizados como material de referencia. Tampoco tienen por qué ser libros de texto.</p>	
Solución	<p>Se debe organizar el curso para introducir los temas a los estudiantes sin abarcarlos por completo, para poder introducir que algunos temas con el nivel básico necesario para poder usarlo en la solución de problemas. Esto permite a los alumnos trabajar sobre problemas de interés sin conocer todas las herramientas que pudieran utilizarse para resolver estos problemas. El instructor puede volver a cada tema, las veces que estime, ofreciendo más información para dominar los temas.</p> <p>En cada ciclo de la espiral, los temas se analizan con mayor profundidad y se incluyen otros temas relacionados. La secuenciación de los "fragmentos" se hace con la intención de proporcionar a los estudiantes habilidades para resolver problemas. El patrón <i>Mix New and Old</i> de Anthony, propone combinar material nuevo con material ya conocido.</p> <p>Los cursos se desarrollan en torno a un tema dado, que es tratado cada vez con mayor profundidad.</p>	
Discusión / Consecuencias / Implementación	<p>Este patrón ofrece mejores resultados en cursos donde un tema puede de ser introducido por etapas que permitan, a cada iteración, resolver problemas cada vez más complejos. Por ejemplo, en informática el concepto de "iteración" es un gran tema. La sentencia "while" es un tema pequeño, sobre todo si se comienza con un bucle simple. Sin embargo, el bucle "while" permite realizar construcciones en programas antes de entender las iteraciones por completo.</p> <p>Para empezar, el profesor debe extraer un subconjunto de materiales que abarquen varios temas que</p>	

Nombre del Patrón:	SPIRAL	#2
<p>interactúen entre sí. En un principio, se deben introducir los casos más simples. El profesor y los alumnos se mueven rápidamente a otros temas hasta comprender como interactúan estos temas. Los alumnos pueden trabajar con el subconjunto de herramientas en la resolución de problemas. Posteriormente, los alumnos trabajaran en un conjunto más amplio de problemas. Esto se puede repetir cuantas veces sea necesario.</p> <p>Los alumnos deben antes tener una idea de cómo encajan las piezas. Una de las consecuencias potencialmente negativas para algunos estudiantes, puede ser que algunas de sus preguntas (¿Qué pasa si ...?) no tengan aún respuesta (ver <i>Test Tube</i>).</p>		
<p>Requisitos</p> <p>El instructor necesita una guía, que muestra el orden en que los temas se presentarán y lo que se aplazará para los ciclos posteriores.</p> <p>El instructor debe formar subconjuntos de cada uno de los muchos temas en los que las herramientas introducidas puedan servir en la solución de problemas. Se deben diseñar subgrupos cada vez más grandes. Se deben diseñar ejercicios para cada uno de los subgrupos. Una forma de diseñar los subconjuntos es a partir de un problema y extraer un conjunto mínimo de herramientas necesarias para resolver ese problema y problemas similares. El subgrupo inmediatamente superior a menudo puede ser diseñado pensando en cómo el problema original se podría ampliar y generalizar su solución.</p>		
<p>Patrones relacionados</p> <p><i>Early Bird</i> puede empezar en el primer ciclo.</p> <p><i>Test Tube</i> se puede utilizar para evitar liarse.</p> <p><i>Toy Box</i> puede ser utilizado para proporcionar una secuencia de ejercicios cada vez más compleja.</p> <p><i>Lay of the Land</i> y <i>Larger Than Life</i> se pueden utilizar para proporcionar una visión global.</p> <p><i>Fixer Upper</i> puede ser utilizado para introducir material nuevo para cada ciclo.</p> <p>Este patrón permite combinar <i>Early Bird</i> y <i>Lay of the Land</i>.</p> <p>Dana Anthony tiene varios patrones relacionados con éste. En particular, <i>Mix New and Old</i>, <i>Seven Parts</i>, <i>Example Lasts One Week</i> y <i>Visible Checklist</i> que se puede emplear para diseñar cada ciclo en espiral.</p>		
<p>Ejemplos</p> <p>Este patrón puede ser usado para enseñar la programación estructurada de bajo nivel. Por ejemplo, se pueden introducir las sentencias de asignación y las formas simples de “if” y “while”. La resolución de problemas usando estos temas permite a los estudiantes acometer problemas simples de programación. En el segundo ciclo, se puede tratar más sobre cada una de estas temas (las sentencias “else” y los bucles infinitos, ...).</p> <p>En el análisis y diseño, se pueden introducir técnicas y herramientas de análisis sencillas (casos de uso simples) y luego la clase puede derivar a diseños simples (tarjetas CRC). Así, los estudiantes pueden tener una idea de todo el proceso, resolviendo problemas sencillos. El segundo ciclo se puede introducir características simples de herramientas más sofisticadas, así como plantear problemas algo más complejos.</p> <p>En los cursos de tecnología de objetos, la herencia simple puede ser introducida y utilizada antes discutir sobre polimorfismo (enlace dinámico).</p>		

Nombre del Patrón:	SPIRAL	#2
Contraindicaciones	Este patrón no puede ser utilizado a pequeña escala. Si esto no es así es mejor evitar emplearlo.	
Referencias	<p>El libro Ten Statement Fortran Plus Fortran IV de Michael Kennedy tuvo un enfoque en espiral.</p> <p>Karel The Robot, de Richard Pattis (Wiley, 1981) planteó una primera aproximación a Pascal con esta orientación. Su sucesor, Karel ++, de Bergin, Stehlik, Roberts, y Pattis (Wiley, 1997) trata de hacer lo mismo con un lenguaje orientado a objetos como C ++ o Java.</p> <p>Los patrones de Dana Anthony se presentaron en PLoP '95.</p>	
Herramientas LMS propuestas	<ul style="list-style-type: none"> • I.9 - Wikis (Glosario) • I.4 - Intercambio de archivos (programa y cronograma) 	
Propuesta utilización	<p>Se dispondrá de un glosario con los términos comunes donde el alumno pueda ampliar información. Si se quiere que todos los alumnos participen en la elaboración de este glosario puede utilizarse wikis. Realizaremos documentos breves con continuos hipervínculos entre ellos, y por esto es necesario definir un guión del curso de forma que el alumno no se disperse. Para algunas asignaturas puede ser necesario realizar constantes ejercicios de evaluación y/o autoevaluación para ir asimilando los temas.</p>	

Nombre del Patrón:	CONSISTENT METAPHOR	#3
Descripción	<p>Cuando vamos a enseñar un tema complejo que no está dentro la experiencia del alumno, podemos hacer uso de una metáfora compleja y consistente para introducir el tema. El alumno necesita conocer las bases de esta metáfora.</p>	
Problema	<p>Para alumnos noveles, es fácil perderse en detalles dentro de un tema. Los alumnos pueden no apreciar como el tema se relaciona con el objetivo del curso. También, es difícil para muchos estudiantes, ver como las cosas encajan y hacer predicciones acertadas de como la tecnología “debe” comportarse.</p>	
Audiencia/Contexto	<p>Se está enseñando a un tema complejo, como pensar en programación orientada a objetos. El tema tiene muchas partes, algunas de las cuales son bastante detalladas. Los estudiantes necesitan una manera de pensar sobre el tema en su conjunto, pero el tema es muy técnico y fuera de su experiencia.</p>	
Fuerzas	<p>Se desea dar a los estudiantes un ejemplo potente y consistente para pensar en algún tema complejo. El ejemplo debe relacionar el tema que enseña con algo dentro de su experiencia. Los estudiantes pueden perderse en los detalles con facilidad y dejar de ver una visión global y cómo las partes se relacionan entre sí. Esto es especialmente cierto cuando los detalles son desconocidos y nuevos para ellos. Es útil cuando se está aprendiendo temas nuevos que relacionan las ideas nuevas ideas con las ya familiarizadas. Se puede precisar mucho tiempo para enseñar todos los elementos sobre el tema en revisión (ver en <i>spiral</i>). Desea que los alumnos se formen una idea acerca de lo que está sucediendo dentro de un sistema y que les permita hacer inferencias válidas sobre lo que debería ocurrir.</p>	
Solución	<p>Crear una metáfora que esté en consonancia con el tema a tratar, y con los mismos elementos básicos que interactúan en la misma forma. Ofrecer esto a los estudiantes como forma de meditar sobre el tema. La metáfora debe permitir a los estudiantes alcanzar conclusiones válidas sobre el tema tratado.</p>	
Discusión / Consecuencias / Implementación	<p>El instructor debe conocer los límites de la metáfora comunicárselas al alumno, para no provocar conclusiones inadecuadas. Una metáfora puede ser utilizada en un pequeño elemento de un tema o para dar una visión de conjunto. Es muy útil cuando se logran conclusiones válidas de ella. Las bases de la metáfora misma debe ser bien conocidas por los estudiantes.</p>	
Requisitos	<p>Los grupos de discusión son útiles en la exploración de metáforas entre los educadores y para encontrar los límites de las metáforas.</p>	
Patrones relacionados	<p>El patrón de Bergin <i>Audio Object Analogies</i> (#31) es un refinamiento de este patrón. <i>Role Playing</i> (#5), también de Bergin, hace uso de este patrón.</p>	
Ejemplos	<p>(Vehículo de tracción humana) Una de las maneras más eficaces para enseñar sobre objetos y sus propiedades es imaginar que cada objeto en un sistema es una persona. La metáfora comparte lo siguiente con los objetos: Las personas y los objetos son actores autónomos. Las personas y los objetos responden a los mensajes. La forma de gestionar una persona u objeto un mensaje, depende del receptor y no del remitente del</p>	

Nombre del Patrón:	CONSISTENT METAPHOR	#3
<p>mensaje.</p> <p>Hay diferentes tipos de personas y objetos (médicos, abogados ...). Sólo porque una persona u objeto responda a los mensajes de igual manera que otra persona u objeto, no significa que necesariamente sean mismo tipo.</p> <p>Diferentes personas u objetos del mismo tipo responden a un mensaje dado de la misma manera.</p> <p>Las personas y los objetos tienen un "estado" que puede cambiar con el tiempo y puede afectar a cómo se responde a un mensaje.</p> <p>La única manera de conseguir que una persona u objeto realicen una acción es solicitándoselo (enviar mensaje).</p> <p>Una persona u objeto puede recibir un mensaje solicitando algún servicio y a su vez puede pedir ayuda a otra persona / objeto, de manera que un "servidor" en una sola transacción puede llegar a ser un "cliente" en otra.</p> <p>No obstante, la metáfora se comporta manera no concurrente cuando los objetos interactúan de forma secuencial, mientras que la gente siempre interactúa al mismo tiempo.</p> <p>(Caja blanca) en la construcción de sistemas, a veces es posible construirlos de manera que los componentes del sistema de información gestionen directamente los componentes del modelo de negocio. En este caso, el sistema del mundo real se convierte en una metáfora para el sistema de información se está construyendo. Aunque este no puede ser el mecanismo dominante en la construcción sistemas orientado a objetos de hoy, sigue siendo útil para noveles que necesitan ayuda inicial en la búsqueda de los objetos del sistema que modela.</p> <p>(CPU) Los modernos sistemas informáticos raramente son exactamente igual que el modelo de memoria que muchos instructores dan a sus estudiantes a pensar sobre la naturaleza de la computación en lenguajes imperativos. El modelo simple, sin embargo, normalmente es suficiente para la mayoría de los propósitos. Es importante tener en cuenta que se trata de una metáfora abstracta.</p>		
Contraindicaciones		
Referencias		
Herramientas LMS propuestas		
<ul style="list-style-type: none"> • I.4 - Intercambio de archivos • III.2 - Autovaloraciones • I.10 - Encuestas • I.3 - e-portafolio • IV.4 - Auditoría 		
Propuesta utilización		
<p>Necesitamos compartir con el alumno la metáfora para que sea entendida por todos. Cualquier duda podrá ser aclarada por foros de consulta. El profesor proporcionará constantes ejercicios de autoevaluación que permitan al alumno practicar con la metáfora propuesta. Al final, se le plantean al alumnos encuestas para determinar su comprensión y la validez de la metáfora seleccionada. Este patrón está especialmente indicado para estudiantes primerizos, y por ello necesitará una mayor supervisión del alumno por parte del profesor (portafolio y auditoría).</p>		

Nombre del Patrón:	TOY BOX	#4
Descripción	La intención de este patrón es dar a los estudiantes un amplio conocimiento histórico y tecnológico del campo, dejándolo "jugar" con herramientas pedagógicas ilustrativas.	
Problema	Con frecuencia, los estudiantes no tienen una percepción real de toda la informática o de sus fundamentos teóricos básicos, tales como máquinas de Turing. Puesto que tenemos que enseñar un montón de cosas, es difícil detenerse en conceptos si deseamos ofrecer la profundidad que se requiere para su comprensión.	
Audiencia/Contexto	El patrón puede ser utilizado en varios cursos y a distintos niveles. Se puede utilizar al principio en los cursos de programación y también en niveles superiores.	
Fuerzas	<p>Los estudiantes deben hacer frente a una gran cantidad de detalles. A veces, no se aprecia la forma en que el detalle se va a utilizar.</p> <p>Muchas aplicaciones de informática son muy complejas y fuera del nivel de habilidad de los principiantes para entenderlas por completo.</p> <p>Mientras se enseña programación, es bueno permitir a los estudiantes programar con las cosas que les muestre algunas ideas simples que se encontrarán más adelante en el plan de estudios.</p> <p>Tenemos que administrar nuestro tiempo con los estudiantes de manera lo más eficiente posible. Por otro lado, los estudiantes necesitan trabajar en problemas.</p>	
Solución	<p>Preparar esqueletos de aplicaciones, cada una de ellas de un área clave, tales como bases de datos u hojas de cálculo. Cada esqueleto constituye el marco para afrontar las actividades y ejercicios de los estudiantes. Si bien el marco está pensado de la manera más sencilla, este puede llegar a ser mucho más complejo si se aprovechan las lecciones aprendidas con anterioridad.</p> <p>Los estudiantes se examinan e interactúan con estos ejemplos escritos especialmente, a escala, de las aplicaciones reales, tales como procesadores de texto, sistemas de bases de datos y hojas de cálculo. Estas aplicaciones se reducen a la forma más simple posible.</p> <p>Elegir ejercicios para ofrecer a los estudiantes un amplio conjunto de experiencias acerca de lo que se puede hacer y lo que es importante en la informática. Estos ejercicios se apoyan en una biblioteca de materiales didácticos, haciendo que el aprendizaje sea divertido. En los primeros cursos, se puede introducir autómatas finitos y máquinas de Turing.</p> <p>Ofrecer a los estudiantes una librería de "clases" que puedan ser utilizadas para implementar algunas funciones complejas. Estos son los bloques de construcción. A continuación, se utilizan para construir un artefacto. Es necesario preparar unas clases previas para realizar esto.</p> <p>Distribuir una librería de "clases" que implementa los conceptos básicos de algunas funciones. Los estudiantes pueden recurrir a estas sin cambios o ampliarlas, e incluso proponer un proyecto.</p>	
Discusión / Consecuencias / Implementación	<p>Es especialmente útil en los cursos de programación orientada a objetos, ya que las clases proporcionadas por el instructor pueden y deben ser utilizadas por los estudiantes en cualquier caso. Pueden ser elegidas cuidadosamente para emular sistemas grandes.</p> <p>Este modelo permite a los estudiantes trabajar activamente con programas más grandes de lo que podrían desarrollar ellos mismos por completo. Si las herramientas se eligen correctamente, también pueden ganar amplitud de comprensión de campo. Si la jerarquía de clases está bien construida, también sirve como modelo para los estudiantes para la construcción de sus propias clases y jerarquías, aunque la intención no es enseñar programación orientada a objetos.</p>	
Requisitos	El instructor debe suministrar una clara jerarquía de clases para implementar las herramientas. Este es un	

Nombre del Patrón:	TOY BOX	#4
trabajo duro, pero se puede compartir los recursos.		
Patrones relacionados		
<p>Este patrón puede proporcionar ejemplos en los ciclos del patrón <i>spiral</i>. Una conjunto de ellos puede apoyar <i>Larger than life</i>, especialmente si están relacionados entre sí. Una conjunto bien escogido también serviría para <i>Lay of the Land</i>. <i>InLook-Outlook</i> puede ayudarnos a decidir qué ejemplos necesitamos.</p>		
Ejemplos		
<p>Un conjunto de clases que implementen puertas lógicas y los circuitos. Las puertas pueden ser conectadas entre sí para formar circuitos. Un conjunto de clases que permita a los estudiantes experimentar fácilmente con los lectores / escritores de los conflictos, por ejemplo. Un juego simple con información completa, que juega contra otro usuario, pero aprende de sus errores. Una programa que simule una hoja de cálculo simple. El programa puede almacenar "programas" en las células utilizando el lenguaje simple. Un conjunto de clases que implementan una base de datos relacional simple que se permita realizar consultas con un lenguaje sencillo. Algunas de estas cosas son complejas, pero con una formación apropiada, incluso los principiantes pueden utilizar y ampliar las herramientas de forma adecuada. Máquina Analítica Decker y Hirschfield es un buen uso de este patrón.</p>		
Contraindicaciones		
Requiere un tiempo de espera bastante largo y un gran esfuerzo por parte de los instructores en la creación búsqueda de materiales. No trate de utilizar este modelo si no dispone de tiempo y recursos.		
Referencias		
La Máquina Analítica: Introducción a la Informática El uso de Internet, Rick Decker, Stuart Hirshfield, PWS, 1998		
Herramientas LMS propuestas		
<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios (almacén de recursos educativos) • V.4 - Herramientas de calificación en línea • I.4 - Intercambio de archivos • I.1 - Foros 		
Propuesta utilización		
Debemos proporcionar el software necesario para practicar los conocimientos adquiridos junto a los manuales de cómo utilizar este software y los ejercicios que proponemos. Puede ser útil disponer de foros que permitan interactuar a los alumnos resolver dudas puntuales (Tipo FAQ). El alumno puede descargar todo el software, librerías u otros objetos que le permitan practicar sin necesidad de estar conectado a la red. El profesor debe poder conocer, incluso evaluar, las experiencias y trabajos realizados por el alumno para tener en cuenta la comprensión de los temas.		

Nombre del Patrón:	TOOL BOX	#5
Descripción	<p>La intención es permitir que los estudiantes construyan una caja de herramientas en los primeros cursos para su uso en cursos posteriores. Si las herramientas están bien pensadas e implementadas, pueden ser una guía maravillosa de software reutilizable. Los estudiantes se convierten en aprendices en el mismo sentido que la gente joven sirvieron como aprendices en los gremios medievales. Allí pasaban sus primeros años con la creación de herramientas que necesitarían si se trataban de alcanzar un grado de maestro en el gremio.</p> <p>Los estudiantes construyen cosas en los primeros cursos que usaran más adelante en el mismo curso y en cursos posteriores.</p>	
Problema	<p>Con demasiada frecuencia, los estudiantes trabajan en un problema y escriben un programa para resolverlo, después descartan el código y trabajan en otros problemas. Muchas veces los estudiantes terminan re-escribiendo el mismo código. Esto es conveniente si se aplica ideas clave, pero a menudo es sólo trabajo perdido que resta tiempo para discutir esas ideas clave.</p>	
Audiencia/Contexto	<p>El modelo se utiliza más intensamente en los primeros cursos de programación, especialmente el curso de estructuras de datos. Sin embargo, se puede utilizar en cursos posteriores, tales como bases de datos, inteligencia artificial, sistemas operativos, compiladores, etc.</p>	
Fuerzas	<p>Los estudiantes en cursos de programación hacen uso del conocimiento de cursos anteriores, pero hacen poco uso de los programas reales construidos antaño. Esto les enseña que las cosas se están continuamente reconstruyendo en los proyectos de software. En realidad, un profesional raramente construye una clase “pila”, en su lugar extrae una de su caja de herramientas y hace uso de ella.</p> <p>Tener un kit de herramientas personales de los componentes de software reutilizables pueden ser de gran ayuda en un proyecto difícil.</p> <p>La construcción de una caja de herramientas ensaya diferentes habilidades. La caja de herramientas debe mantenerse. Sus elementos deben construirse pensando en la generalidad. El encapsulado es lo más importante en una caja de herramientas.</p>	
Solución	<p>Los ejercicios para los estudiantes tienen varias partes. Una parte de cada ejercicio es construir una herramienta general que pueda ser útil en otros proyectos y dedicar algo de esfuerzo en su formulación adecuada para su reutilización. El diseño para la reutilización debe ser explícito y debe discutirse entre los alumnos y comentarse por el profesor. Distintos grupos de estudiantes pueden combinar los diseños individuales de las herramientas, discutir los méritos relativos de cada uno y construir una aplicación común que mejore cada uno de los diseños individuales.</p>	
Discusión / Consecuencias / Implementación	<p>Los estudiantes adquieren habilidades, en los primeros cursos, en la construcción de componentes reutilizables. En la medida en que fallen, se refuerzan los temas en cursos posteriores, necesitando reconstruir las partes de la caja de herramientas para su uso en proyectos. Es necesario utilizar lenguajes que permitan la reusabilidad. La mayoría de estos lenguajes están orientados a objetos o son funcionales. Scheme, C++, Eiffel, Java, Ada son buenas opciones. Hay algunos otros, tales como Modula 3, Beta, Oberon, CLOS, y el Standard ML. Son especialmente útiles los lenguajes que soporten “templates”.</p> <p>Los profesores pueden proporcionar algunas herramientas (estructuras de datos y algoritmos) en forma de jerarquías de clase o en otras librerías. Los estudiantes completarán esta colección. Se debe destinar tiempo a evaluar la construcción de herramientas y su corrección, pero también su capacidad de reutilización. Los instructores en cursos posteriores deben ser conscientes de que los estudiantes tienen estos kits personales de herramientas que deben usar en los ejercicios y proyectos, así como permitir su ampliación.</p>	

Nombre del Patrón:	TOOL BOX	#5
Un elemento importante para este modelo es la coordinación con los cursos posteriores.		
Requisitos	<p>Es necesaria una reflexión en el diseño en los cursos iniciales, en cuanto a qué herramientas son útiles en términos generales, pero sobre todo qué herramientas serán necesariamente útiles en cursos posteriores. Debe considerarse la independencia de la plataforma. Es necesario un plan general y se debe proporcionar implementaciones de estas herramientas que los estudiantes necesiten, pero no vayan a construir.</p>	
Patrones relacionados	<p><i>Fixer Upper</i> puede ser utilizado por el instructor para proporcionar herramientas parcialmente incompletas o erróneas para que los alumnos las completen o reparen. La reparación o refinado de herramientas se puede hacer pensando en <i>spiral</i>.</p>	
Ejemplos	<p>Las herramientas en el libro de software de Kernighan y Plauger eran un buen uso de este patrón. Probablemente este libro debería ser leído por cualquier persona que quisiera poner en práctica este modelo. Se muestra cómo un alto grado de reutilización se puede lograr con herramientas muy simples. Muchas estructuras de datos, con sólo una ligera modificación, podrían usarse como ejemplos de este patrón. Una clase de “pila” se construye normalmente al principio. No se necesita una gran cantidad de trabajo extra para su construcción original, de tal manera que suele ser útil en muchos proyectos. Lisp, ML, y Scheme dependerá en gran medida de las librerías de funciones. Estas pueden construirse en los primeros cursos y ser utilizadas posteriormente.</p>	
Contraindicaciones	Este patrón probablemente no deba utilizarse en cursos de teoría o “conceptos”.	
Referencias	Software Tools in Pascal, Brian Kernighan, PJ Plauger, Addison Wesley, 1981	
Herramientas LMS propuestas	<ul style="list-style-type: none"> • I.3 - e-portafolio • I.8 - Blog • VII.3 - Librerías digitales o repositorios • III.1 – Grupos 	
Propuesta utilización	<p>Deben conservarse las herramientas de los alumnos para su uso más adelante. Seguramente este material podrá ser compartido por otros usuarios del mismo u otro grupo. El alumno puede compartir su trabajo en su portafolio, junto a un blog o foro que ayude a documentar los problemas/avances con dichas herramientas. Para poder hacer uso en cursos posteriores de este material, puede alojarse en una librería digital. Se obtendrán mejores resultados si acostumbramos al alumno a trabajar en grupo.</p>	

Nombre del Patrón:	LAY OF THE LAND	#6
Descripción	A los estudiantes se les muestra, de forma temprana, un “artefacto” grande para su observación, con la intención de mostrar la complejidad del campo que están a punto de estudiar.	
Problema	A menudo se imparten cursos que cubren mucho terreno. Si los estudiantes no reciben una visión global al principio, puede que nunca lo entiendan, y mientras se pierdan en un mar de detalles. Sería deseable mostrar a los alumnos la amplitud de un gran “artefacto”, para que tengan con que relacionar lo que aprenden y no perderse en los detalles a medida que avanzan.	
Audiencia/Contexto	Esto tiene una aplicación muy amplia en casi todos los ámbitos. Es especialmente útil en la enseñanza de temas con una gran cantidad de piezas que deben encajar en ciertos aspectos. La enseñanza de la programación es un ejemplo. Diseño metodológico es otro.	
Fuerzas	<p>La enseñanza es a menudo gradual, con temas introducidos uno tras otro. Esta puede no ser la mejor manera de aprender, sobre todo si se trata de imponer una disciplina estricta en el orden de las ideas. La teoría constructivista de la educación está en desacuerdo con esta propuesta pues puede no ser útil para el estudiante mostrar el sistema antes de construirlo. La mente no es una tabla rasa en la que se vierte ideas. Los estudiantes necesitan ver la visión global, así como el detalle.</p> <p>Al principio, sólo pueden producir artefactos sencillos, pero se puede examinar, aunque sólo sea de manera superficial, un artefacto complejo. La mayoría de la gente puede leer y entender algo mucho más complejo de lo que puede producir por sí mismo.</p> <p>Tener una visión global puede incentivar el estudio de las partes, ya que tienen una idea de cómo podrían ser utilizadas. Esto es especialmente cierto si puede acompañarse con un ejemplo que sea de interés.</p>	
Solución	Dar a los estudiantes un gran artefacto para ser examinado al principio del curso. Se puede ver qué es lo que se supone que se tratará en este curso y qué tipo de cosas se espera que lleguen a dominar. El artefacto debe tener la complejidad de algo que gustaría que fueran capaces de construir al final. Dedicar tiempo a examinar las partes y sus interacciones. El artefacto debe incluir la mayor parte de los elementos que se estudiarán en ese curso. Es bueno que el artefacto tenga algunos puntos sutiles. Si surgen preguntas inicialmente sobre estos puntos pueden ser postergadas, pero puede significar que se tiene mejores estudiantes de lo que se piensa. Retomar el artefacto durante todo el curso y revelar y discutir sus puntos sutiles.	
Discusión / Consecuencias / Implementación	<p>Los estudiantes tienen la oportunidad de ver el objetivo de su estudio. También tienen un modelo en el que pueden basar su propio trabajo.</p> <p>El artefacto debe estar preparado antes de tiempo. Los proyectos de alumnos de años anteriores son una buena fuente, aunque quizás deban modificarse un poco para destacar puntos que parezcan más importantes. Si los artefactos son grandes, una buena manera de transmitirlos a los estudiantes es a través de Internet, sobre todo en la Web. De esta manera, otros educadores podrán utilizarlos también.</p>	
Requisitos	Se necesitan buenas soluciones para proyectos más grandes. Estas deben exhibir una excelente estructura y estilo, pues se espera que los estudiantes emulen su estructura y estilo.	
Patrones relacionados	Uno de ellos podría ser fácilmente <i>Larger than Life</i> . <i>Spiral</i> podría utilizarse para examinar piezas del artefacto en ciclos sucesivos. Esto es recomendable si se complementa muy bien las piezas. El artefacto también podría ser un <i>Fixer Upper</i> . Si ha seleccionado cuidadosamente, es también <i>Early Bird</i> . <i>Test Tube</i> puede utilizarse para permitir que los alumnos respondan a preguntas sobre el artefacto.	
Ejemplos		

Nombre del Patrón:	LAY OF THE LAND	#6
<p>Un gran programa orientado a objetos con algunas clases interactuando de una manera interesante es una buena opción en un primer curso de programación utilizando un lenguaje orientado a objetos. Un diseño completo con muchos documentos puede utilizarse en un curso de diseño de sistemas. Un diseño complejo de bases de datos de entidad relación gráficos y tablas puede utilizarse en un curso de diseño de base de datos. Algunos instructores del curso de compiladores dan a los estudiantes un compilador completado para un lenguaje y, a continuación, les solicitan que proporcionen un compilador para otro en durante el curso.</p>		
Contraindicaciones		
Referencias		
Herramientas LMS propuestas	<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios • I.4 - Intercambio de archivos (programa) • I.10 - Encuestas 	
Propuesta utilización	<p>Los escenarios a mostrar pueden ser comunes a otras asignaturas o particulares como prácticas de cursos pasados. Se deben proporcionar el programa de la asignatura de modo que el alumno conozca el progreso en cada momento. Por otro lado, se pueden proporcionar encuestas al final del curso que permitan conocer como de atractivo resultado el programa y si este se cumplió satisfactoriamente.</p>	

Nombre del Patrón:	FIXER UPPER	#7
Descripción	Se ofrece al estudiante o un grupo de estudiantes un gran artefacto que es generalmente conocido, pero con defectos cuidadosamente introducidos. Servirá tanto para introducir un tema complejo temprano, como para introducir la corrección y análisis de errores. A los estudiantes se les pide que reparen y discutan el artefacto.	
Problema	Demasiado a menudo los estudiantes trabajan en problemas "de juguete" porque no pueden tener la experiencia o la habilidad para construir grandes artefactos desde cero y no disponen de tanto tiempo. Pero todos los problemas reales son grandes y ya ha pasado el día en que los pequeños problemas eran interesantes. Se desea que los estudiantes trabajen en grandes artefactos sin abrumarse. Por otro lado, los estudiantes también tienen dificultad cuando surgen los errores inesperados en su propio trabajo. Por ejemplo, los mensajes de error de compiladores y en tiempo de ejecución, a menudo les dejan perdidos.	
Audiencia/Contexto	El patrón puede utilizarse en varios cursos y a distintos niveles. Pueden utilizarse muy pronto en los cursos de programación y en los de análisis y diseño. También puede utilizarse para mostrar la estructura general de una solución metodológica.	
Fuerzas	A menudo se tiene que introducir al alumno en un nuevo campo que requiere el dominio de varios temas. Los estudiantes a menudo no logran ver cómo los temas encajan entre sí cuando se introducen secuencialmente. Con frecuencia, tampoco tienen una comprensión de los medios para localizar y corregir errores. Preparar un artefacto más grande que pueden ser creado por los estudiantes esta generalmente a su alcance. Les da una mejor idea de la proporción de problemas interesantes y les permite integrar una serie de cuestiones en la solución de un problema individual. Los estudiantes pueden beneficiarse descubriendo problemas más grandes que puede resolver en su estado actual de desarrollo. También necesitan habilidades de análisis crítico y la capacidad para evaluar programas, dibujos, etc. (véase <i>Lay of the Land y Larger than Life</i>).	
Solución	Dar a los estudiantes un artefacto, como un programa o diseño. Se propone el artefacto como solución a un problema, pero mientras el instructor ha introducido deliberadamente defectos en el programa, diseño o lo que sea. El artefacto debe ser bastante grande y debe contener un número de defectos. La mayoría de los defectos debe ser simples y claros para la mayoría de los lectores. Debe haber uno o dos defectos más profundos. Solicitar a los estudiantes encontrar y corregir los defectos. Se les debe pedir que discutan la naturaleza de los defectos encontrados y las razones de sus cambios. Por último, se les debe pedir que discutan la estructura general del artefacto y extraer conclusiones.	
Discusión / Consecuencias / Implementación	Este modelo permite a los estudiantes a trabajar activamente con artefactos más grandes que puedan desarrollar completamente ellos mismos. Se benefician encontrando fallos en su propio trabajo, que es una habilidad valiosa. En la programación, los estudiantes ven errores léxicos, sintácticos y semánticos. En diseño, pueden ver el efecto de un incorrecto reparto de responsabilidades. Es importante que la estructura general del artefacto sea conocida. Si se trata de un programa debe ser bien diseñado y escrito, con una buena elección de identificadores. Si es un documento de análisis o diseño, su estructura debe ser sólida con un mapa claro en el planteamiento del problema. La mejor manera de desarrollar tal artefacto es comenzar con una excelente solución para un problema y, a continuación, debe proponer defectos. Debe haber diferentes tipos de defectos, no defectos estructurales si se trata de novatos. Esta última regla puede romperse si el artefacto se presenta al final del curso, en lugar de al principio.	
Requisitos	Un problema y artefacto bien diseñado e implementado que se pueda manipular para proporcionar los errores necesarios. Estos pueden provenir de libros de diseño de calidad industrial y profesional.	

Nombre del Patrón:	FIXER UPPER	#7
También puede provenir de los proyectos de años anteriores, siempre que está dispuesto a modificar y mejorar el artefacto para que realmente sea de alta calidad antes de la introducción de defectos.		
<p>Patrones relacionados</p> <p>Cuando el artefacto se presenta al comienzo del curso y presenta las ideas clave del curso, entonces es también <i>Early Bird</i>. Cuando el artefacto es muy grande, entonces es también que una instancia <i>Larger than Life</i>. Si el artefacto también es ilustrativo de algún tema importante, entonces también es una instancia de <i>Toy Box</i>. Cuando el artefacto ilustra el tema general de interés en el curso, se relaciona con <i>Lay of the Land</i>. Si el artefacto reparado es útil en algún contexto también puede ser una instancia de la <i>Tool Box</i>. Este tiene algunos de los mismos objetivos que <i>Mistake</i>, aunque es abordado de forma diferente. Aquí nos encontramos con errores. En <i>Mistake</i> hacemos errores específicos. Este patrón es una forma de lograr la comprensión lectora antes de escribir.</p>		
<p>Ejemplos</p> <p>Este patrón se ha utilizado para enseñar:</p> <ol style="list-style-type: none"> 1. Introducción a la programación. Aquí el artefacto es un programa que ilustra una serie de construcciones sintácticas que aún no se han introducido en clase. (Se ha utilizado como el primer ejercicio). El programa puede ser lo suficientemente grande como para que su estructura no sea obvia. Dos o tres clases con varios métodos cortos que cada uno será suficiente. Una parte del programa podría ser más compleja. Junto con el “driver”, debe haber tres o más páginas de código. Los errores pueden ser de mayoría sintáctica y léxica, para que el compilador pueda encontrarlos. Uno o dos errores semánticos también deben introducirse, por lo que el programa no funciona según lo previsto. Más graves, y tal vez para estudiantes más avanzados, sería incumplimiento de una condición previa pactada. De diez a quince errores serán suficientes si la mayoría son percibidos fácilmente. Puede introducirse incluso una sola clase que cuenta con una interfaz pública defectuosa. Se solicitará a los estudiantes analizar las consecuencias de esto en relación con el probable uso futuro de la clase. 2. Introducción al diseño. Aquí se presenta un problema y un diseño de la solución. De seis a diez elementos importantes en el diseño serán suficientes. El diseño debe tener unos fallos simples, tales como falta de rutas de mensaje o funcionalidad que falta dentro de una clase. Un reparto incorrecto de responsabilidades en una pequeña región del diseño es relativamente fácil introducir. Un subsistema podría quedar afuera. Si el diseño requiere varios documentos, puede haber una incoherencia entre los documentos. Cinco o seis fallos es suficientemente. 3. Algunos (Linda Rising) han utilizado este patrón con artefactos muy grandes. Lo suficientemente grandes como para que no se entiende por una sola persona en el nivel de los estudiantes y por lo tanto requieren trabajo en equipo. Ver que <i>Groups Work</i>. 		

Nombre del Patrón:	FIXER UPPER	#7
Contraindicaciones	<p>Debe utilizarse cuidadosamente si la honestidad del estudiante es un problema. Es fácil para un estudiante localizar errores en los programas de C++, por ejemplo. Una forma de solucionar esto es utilizar artefactos grandes que requieren trabajo en equipo. Otra es hacer preguntas relativas a la estructura, así como de los errores. Por último, podemos solicitar a los estudiantes examinar el artefacto antes de que se les proponga un conjunto de preguntas acerca de él. Algunos estudiantes se frustran ante artefactos demasiado grandes. El instructor debe estar preparado para proporcionar apoyo y aliento tal como en el mundo real, por dos motivos: (a) se sientan frustrados y (b) carecen de conocimientos.</p>	
Referencias	<p>Kyle Brown (el primer patrón pastor) y los participantes en el OOPSLA '98 Pedagogical Patterns.</p>	
Herramientas LMS propuestas	<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios • V.4 - Herramientas de calificación en línea (tareas) • III.1 - Grupos • I.1 - Foros • I.7 - Servicios de presentación multimedia (pizarra electrónica) 	
Propuesta utilización	<p>El profesor debe tener un conjunto de "proyectos" comunes o particulares (pueden ser trabajos de otros alumnos de cursos pasados) sobre los que introducirá algunos errores para que el alumno o grupo pueda identificarlos. Los errores deben ser simples o más complejos dependiendo del nivel de la asignatura y los alumnos. Se asignarán tareas por un tiempo y se ofrecerán foros por grupo en los que pueda participar el profesor para contestar dudas y ofrecer ayuda. La identificación y solución de los problemas puede ser expuesta en la propia tarea o en una pizarra electrónica.</p>	

Nombre del Patrón:	LARGER THAN LIFE	#8
Descripción	Estudiantes de programación orientada a objetos y diseño pueden y deben examinar y utilizar artefactos computacionales mucho antes de que los pueden diseñar y construir ellos mismos.	
Problema	Los artefactos reales en este campo son grandes y complejos. A menudo, también los estudiantes trabajan solos, cuando en la mayoría de los trabajos reales lo hacen en equipos. Finalmente, mucho trabajo real en el mundo de la informática no es construir un artefacto desde cero, sino para modificar uno ya existente. Los proyectos más reales en el mundo orientado a objetos tienen un ciclo de vida de crecimiento gradual en el que un gran artefacto crece a partir de un núcleo pequeño y bien diseñado.	
Audiencia/Contexto	Estudiantes en todos los niveles, desde novatos a muy experimentados pueden utilizar este patrón. Es un ejemplo de <i>Read Before Write</i>	
Fuerzas	<p>Inicialmente, los estudiantes no pueden producir programas y otros artefactos de desarrollo de software de cualquier complejidad. Sin embargo deben introducirse a la complejidad de sistemas reales. Pueden, de hecho, estudiar y aprender de programas y otros artefactos más allá de su capacidad para producirlos. Se gasta mucho tiempo en clase en niveles de detalle muy bajos, sin embargo, los estudiantes necesitan tener una visión global.</p> <p>Si los estudiantes construyeran todo lo que ellos utilizan, obtendrán una visión errónea de cómo se produce en el mundo el software real. También suelen tener una apreciación cierta de la complejidad de sistemas reales o los requisitos de software que serán usados por personas distintas de su autor.</p> <p>En cursos distintos de informática, los estudiantes trabajan regularmente con textos y otros artefactos mucho mayores y más sofisticados que podrían producir ellos mismos. Un curso de literatura o historia es un buen ejemplo. Lo mismo ocurre en los cursos de sociología. Estos artefactos enseñan a los estudiantes qué es lo mejor en el campo y como deben emularlos.</p>	
Solución	<p>Dar acceso a los estudiantes a diseños y grandes programas antes de que tengan la capacidad para producirlos. Estos artefactos pueden utilizarse como base de ejercicios. Los estudiantes pueden hacer pequeñas modificaciones para programas grandes y pueden ampliarlos desde el principio. También puede realizar correcciones sobre programas grandes defectuosos si los fallos son de dificultad moderada (véase <i>Fixer Upper</i>).</p> <p>Por ejemplo, a los estudiantes se les pide, a menudo, estudiar un compilador para un lenguaje concreto, mientras aprenden de otro compilador para otro lenguaje distinto en el curso de compiladores. Incluso en el primer curso, un diseño de un programa bastante (simulación de supermercado), podría utilizarse para su estudio y como base para ejercicios. Estos artefactos generalmente deben tener una alta calidad de diseño, para que los estudiantes tengan una modelo que emular en su propio trabajo. Una excepción es si el objetivo del ejercicio es mejorar un diseño grande pero pobre.</p>	
Discusión / Consecuencias / Implementación	A diferencia de los días en que los lenguajes eran simples y las librerías inexistentes, los estudiantes necesitan hoy poder utilizar librerías escritas por otros. Estas librerías son generalmente más complejas que las que los estudiantes pudieran crear. También pueden contener mucho código, generalmente de bajo contenido conceptual, por lo que no merece gastar tiempo del curso en desarrollarlos desde cero. Su diseño, sin embargo, puede ser de interés.	
Requisitos	Grandes artefactos computacionales de buen diseño.	
Patrones relacionados	<p>Si el artefacto requiere reparación también es un <i>Fixer Upper</i>.</p> <p>Si incide en las grandes ideas y se ofrecen pronto es un <i>Early Bird</i>.</p>	

Nombre del Patrón:	LARGER THAN LIFE	#8
<p>Si es generalmente útil es una entrada de la <i>Tool Box</i>. Si el artefacto cubre la mayor parte de los principales conceptos en un curso también es una instancia de <i>Lay of the Land</i>.</p>		
Ejemplos	<p>La librería de plantillas estándar de C++ es un artefacto. La documentación de diseño para grandes sistemas también son ejemplos. Los programas completos como sistemas operativos y compiladores también son útiles.</p> <p>Es una costumbre en un curso de compiladores dar a los estudiantes un compilador completado de un lenguaje de programación al comienzo del curso. Este artefacto, a continuación, se examina en un plazo y los estudiantes construyen un compilador para un lenguaje diferente. En cualquier curso orientado a objetos, los estudiantes se beneficiarán del uso de librerías preparadas por otros.</p> <p>Linda Rising ofrece un ejemplo de un proyecto de diseño de software que es <i>Larger than Life</i> y un buen <i>Lay of the Land</i>. Tiene algunas características de un <i>Fixer Upper</i>.</p> <p>Owen Astrachan analiza la idea de un <i>software cadáver</i> que es un artefacto de estudio por los alumnos. A menudo exhibe características de <i>Larger than Life</i>.</p>	
Contraindicaciones		
Referencias	<p>Rising. Removing the Emphasis on Coding in a Course on Software Engineering. SIGCSE BULLETIN. Febrero 1989.</p>	
Herramientas LMS propuestas	<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios (almacén de recursos educativos) • V.4 - Herramientas de calificación en línea • III.1 - Grupos • I.1 – Foros 	
Propuesta utilización	<p>Se le ofrece al alumno un proyecto dentro de un almacén de recursos educativos. El alumno podrá examinarlo y comprobar su funcionamiento. Se le propone como tarea individual o de grupo elaborar partes de ese proyecto con distintas funcionalidades pero donde pueda aprovechar el diseño. Un foro por grupo puede ser necesario para fomentar la colaboración y resolver las dudas.</p>	

Nombre del Patrón:	STUDENT DESIGN SPRINT	#9
Descripción	Los estudiantes necesitan practicar diseño en todos los niveles. Este patrón les permite una rápida revisión entre pares y comentarios desde los primeros momentos.	
Problema	La mayoría de los educadores reconoce que los estudiantes deben presentar sus diseños desde el principio. También reconoce la necesidad de trabajar en equipo y de un análisis crítico. Finalmente es necesario presentar el diseño del sistema, pero alumnos principiantes necesitan el diseño de los programas. Si no se enseña, los estudiantes desarrollarán sus propias técnicas ad-hoc que pueden reforzar malos hábitos.	
Audiencia/Contexto	Este patrón se aplica a cursos para principiantes en programación y diseño. También puede utilizarse con los desarrolladores experimentados que son nuevos en un tema. Consigue aplicar <i>Active Student</i> .	
Fuerzas	Los estudiantes necesitan practicar diseño, tanto diseño de programas como de sistemas. Pero el diseño es duro. Necesitan información desde el principio. Cuanto antes lleguen los comentarios, mejor. Necesitan ver buenos diseños y criticar diseños pobres. Necesitan ver las consecuencias de un diseño pobre.	
Solución	<p>Esta actividad puede tener lugar en el aula o en un laboratorio. Los estudiantes se dividen en grupos de dos (o tres). Se proporciona un problema de diseño y se pide a los equipos que propongan un esquema de diseño en 15 minutos. Debería haber un bosquejo escrito del diseño en ese momento. El instructor puede mirar por encima del hombro y comentar o no, pero deben ofrecerse algunos consejos. Las preguntas deben responderse libremente.</p> <p>Al final de 15 minutos, el instructor plantea un conjunto de preguntas acerca de los diseños sin pedir respuestas. Las preguntas deben ser tales que no puede ser respondidas favorablemente por un conjunto de diseños pobres.</p> <p>Los estudiantes se reagruparon luego combinando pares de grupos cercanos, por lo que ahora tiene grupos de 4 ó 5 alumnos y cada grupo tiene dos de los diseños originales. La tarea ahora es ligeramente modificada y los grupos deben proponer un nuevo diseño.</p> <p>Tras otros 15 minutos, el instructor plantea nuevamente un conjunto de preguntas para ser contestadas, reagrupa los alumnos otra vez en grupos más grandes aún, modifica la tarea ligeramente y, de nuevo, los pone a trabajar.</p> <p>Este proceso puede continuar tantos ciclos como desee el instructor. Al final, el instructor debe evaluar los diseños resultantes y hacer comentarios. Puede ser suficiente para mostrar uno o dos de los mejores diseños y explicar por qué son mejores que los otros. Si existen diseños pobres también deben mostrarse, sin que figuren los nombres de los diseñadores. Por otra parte, los grupos pueden presentar y justificar sus diseños y el resto de la clase puede criticarles.</p>	
Discusión / Consecuencias / Implementación	<p>En algunas situaciones puede ser un ciclo de todo lo que se necesita, seguido de una discusión de los temas. En este caso el instructor puede pedir a los grupos que diseños tenían ciertas características.</p> <p>Si el ejercicio es diseñar un sistema de tarjetas CRC, un buen seguimiento es aplicar el patrón <i>Role Playing</i> o <i>Incremental Role Playing</i>. Se puede pedir a los alumnos con un buen diseño, pero no perfecto, explorar las vías de comunicación de su diseño en busca de “cuellos de botella”. Este patrón no se limita al análisis y diseño. Se puede usar para el diseño de programas y el diseño de estructuras de datos.</p>	
Requisitos		
Patrones relacionados	Esto puede conducir a un patrón de <i>Role Play</i> .	

Nombre del Patrón:	STUDENT DESIGN SPRINT	#9
Ejemplos	<p>Alistair Cockburn tiene un ejercicio maravilloso para los estudiantes de diseño de una máquina de café en tres o cuatro ciclos en que los requisitos se complican cada ciclo. En el primer ciclo de la máquina puede ofrecer café a 35 centavos. En el segundo también puede ofrecer sopa a 25 centavos.</p> <p>Este patrón puede utilizarse en el diseño de programas de cursos de primero y segundo. La tarea puede ser escribir una función con un determinado conjunto de pre y post condiciones. Las tareas en los ciclos posteriores pueden ser endurecer las condiciones pre y/o fortalecer las condiciones de post.</p> <p>Este patrón puede utilizarse en el diseño de estructuras de datos en el curso de la estructura de datos. Por ejemplo, a los estudiantes se les puede pedir diseñar una lista cruzada, sin decirles cómo será utilizada. Se debe diseñar un protocolo y elegir una estrategia de aplicación. A continuación, el instructor puede sugerir algunos usos que podría poner una lista cruzada y preguntar si el diseño es compatible con ese uso.</p>	
Contraindicaciones		
Referencias	<p>“The coffee machine” puede encontrarse en la web de Alistair Cockburn: http://members.aol.com/acockburn/</p> <p>Role Playing, David Belin, http://www-lifia.info.unlp.edu.ar/ppp/pp5.htm</p> <p>Incremental Role Play, Jutta Eckstein, http://www-lifia.info.unlp.edu.ar/ppp/pp7.htm</p>	
Herramientas LMS propuestas	<ul style="list-style-type: none"> • V.4 - Herramientas de calificación en línea • III.1 - Grupos • I.4 - Intercambio de archivos (gestor de documentos) • I.5 - Herramientas de comunicación síncronas (chats/videochats) • I.7 - Servicios de presentación multimedia (pizarra electrónica) 	
Propuesta utilización	<p>Dividir los alumnos en pequeños grupos a los que se propone elaborar un "proyecto" simple. Toda la información que precise podrá obtenerse en un gestor de documentos o enlaces externos. Pasado un tiempo se les plantea algunas preguntas globales a todos los grupos, se seleccionan los mejores proyectos y se reorganizan los grupos en torno a estos proyectos donde, al menos, 2 ó 3 miembros son del grupo anterior. Sobre el proyecto elaborado se le solicita nuevos requisitos y así sucesivamente. Para ganar en rapidez se puede emplear herramientas síncronas como chats o videochats. Cada grupo puede defender su proyecto utilizando herramientas participativas como pizarras electrónicas. Si el proceso durara varios días se podría utilizar foros para debatir los requerimientos solicitados para el proyecto.</p>	

Nombre del Patrón:	MISTAKE	#10
Descripción	Se pide crear a los estudiantes un artefacto como un programa o un diseño que contiene un error específico. El uso de este patrón explícitamente enseña a los alumnos cómo reconocer y corregir errores. Se pide al estudiante explícitamente realizar ciertos errores y, a continuación, examinar las consecuencias.	
Problema	Las personas cometen errores. Los estudiantes a menudo no saben cómo interpretar los mensajes de error proporcionados por sus herramientas o qué hacer para resolver los problemas que son diagnosticadas por las herramientas. La depuración es una habilidad esencial, puede hacerse con un depurador sofisticado o simplemente mediante la comparación de resultados reales con las expectativas. Inicialmente, los estudiantes no saben lo que ocurre cuando los errores se han producido, por lo que no saben qué hacer cuando vean un diagnóstico o resultados incorrecto de un programa.	
Audiencia/Contexto	Es muy aplicable en las primeras etapas del aprendizaje de programación. Los errores sintácticos y semánticos son frecuentes y los estudiantes deben familiarizarse con los mensajes producidos por compiladores, sistemas de tiempo de ejecución sobre el programa. El patrón podría utilizarse también en un curso de análisis o diseño en que ciertos errores específicos, pero bastante comunes, necesarios para formar parte del diseño. El instructor puede proporcionar una parte del diseño, que es erróneo de alguna manera, y se piden a los estudiantes para completar el diseño sin cambiar parte del instructor. (Ver el patrón <i>Fixer Upper</i> .) Este patrón se utiliza a menudo en cursos de base de datos y sistemas operativos donde se piden a los estudiantes explorar las condiciones que conducen a un bucle. Del mismo modo, en el aprendizaje de programación concurrente, a los estudiantes se les pide escribir programas en que las condiciones de carrera entre procesos conducen a la corrupción de datos.	
Fuerzas	Los estudiantes, como los profesionales, cometen errores. Los profesionales, generalmente saben cuáles son las indicaciones de un error, pero no así los estudiantes. Normalmente ayudamos a los estudiantes a evitar errores, pero de todos modos se producen. Los alumnos deben tener una forma de explorar los efectos de los errores para que les pueden reconocer por sus efectos.	
Solución	Pedir a los alumnos que propongan un artefacto con ciertos errores específicos (normalmente un solo error). Luego se explora el efecto del error. Por ejemplo, dar a los estudiantes una asignación en la que se dio instrucciones para crear y ejecutar un programa con ciertos errores específicos. Pedirles que comenten los diagnósticos producidos o no como causa del error.	
Discusión / Consecuencias / Implementación	Los estudiantes deben familiarizarse más con los errores y cómo corregirlos. Habiendo visto errores concretos, pueden aprender a evitarlos desde el primer momento. Deben prepararse, con cuidado, ejercicios utilizando este patrón y asegurarse de resolverlos antes de presentarlos en la clase, con el fin de evitar ocurrencias imprevistas que puedan aparecer en clases. Es importante que los estudiantes tienen la oportunidad de discutir temas interesantes que se producen al hilo de estos errores. Una técnica especialmente útil es solicitar a los alumnos que cometan errores que, cuando se producen accidentalmente, son especialmente difíciles de diagnosticar.	
Requisitos	El instructor simplemente necesita conocimiento de errores comunes.	
Patrones relacionados	En <i>Fixer Upper</i> el instructor hace los errores y los estudiantes los corrigen. En <i>Test Tube</i> solicitamos que lo exploren. Aquí les pedimos exploren únicamente errores concretos.	
Ejemplos		

Nombre del Patrón:	MISTAKE	#10
<p>Además de los mencionados anteriormente, este patrón puede utilizarse eficazmente para enseñar a los estudiantes el uso de punteros en lenguajes como C o C++, y proponer realizar todas las operaciones más comunes. Este uso particular es algo peligroso en equipos que tengan memoria de e/s y sistemas operativos desprotegidos. Los errores sintácticos y semánticos son explorados con este patrón. Un ejercicio de un viejo libro [Teague] fue escribir un programa que produjese cada diagnóstico mencionado en los manuales para un determinado compilador (Fortran).</p>		
Contraindicaciones		
<p>Este patrón puede utilizarse en exceso. Se utiliza mejor al principio y en respuesta a preguntas concretas de los alumnos.</p>		
Referencias		
<p>Computing Problems for Fortran Solution, Robert Teague, Canfield Press, 1972.</p>		
Herramientas LMS propuestas		
<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios (almacén de recursos educativos) • V.4 - Herramientas de calificación en línea • III.1 - Grupos • I.1 - Foros 		
Propuesta utilización		
<p>El alumno podrá utilizar algún "proyecto" privado o público donde se le invitará a realizar algún cambio. Estos proyectos pueden residir en un repositorio. A cada alumno o grupo de alumnos se le asigna una tarea que puede ser debatida y asistida mediante foros moderados por el profesor.</p>		

Nombre del Patrón:	TEST TUBE	#11
Descripción	<p>Los estudiantes pueden responder a preguntas de programación del tipo "Qué pasaría si..." accediendo a un ordenador. A veces es más rápido y más eficaz que la documentación formal. Los estudiantes también pueden explorar las partes indefinidas de algunos lenguajes de programación.</p> <p>También se le conoce por <i>Try It and See</i>.</p>	
Problema	<p>A menudo los estudiantes son ingenuos acerca de qué recursos están disponibles para ellos. A veces realizan preguntas que podrían responder ellos mismos. Aunque esto no es malo, el no hacerlo podría proporcionarles más tiempo para poder encontrar sus propias respuestas rápidamente. Los estudiantes necesitan aprender de forma eficaz para responder a sus propias preguntas.</p>	
Audiencia/Contexto	<p>Cursos de iniciación a la programación.</p>	
Fuerzas	<p>Los estudiantes pueden desaprovechar cantidad de tiempo de clase con preguntas de muy bajo nivel. También se pueden frustrar si no se encuentran documentación suficientemente detallada para responder a sus preguntas.</p>	
Solución	<p>Dar a los alumnos varios ejercicios en los que se les solicita escribir pequeños programas destinados a responder preguntas simples de la forma "Qué pasaría si...". Estos ejercicios deberán realizarse con frecuencia para que los estudiantes se acostumbren a sondear a la máquina, en lugar de la documentación.</p>	
Discusión / Consecuencias / Implementación	<p>Algunos lenguajes de programación están sin definir. En estos lenguajes, el instructor deberá señalar algunos lugares donde no hay reglas. Los estudiantes deben ser conscientes de estas situaciones para que puedan programar conscientemente y no sacar conclusiones engañosas.</p>	
Requisitos	<p>Las preguntas de los alumnos pueden ser una buena fuente para los ejercicios de este tipo. El instructor puede responder la pregunta de una estudiante y convertirlo en un ejercicio de respuesta corta. Si se dispone de un laboratorio, los estudiantes pueden probar hipótesis acerca de cómo funcionan las cosas.</p>	
Patrones relacionados	<p>Puede usarse conjuntamente con <i>Fixer Upper</i>.</p> <p>Si el objetivo del ejercicio es generar errores, esto es una instancia de <i>Mistake</i>. No obstante, este patrón es más general que <i>Mistake</i>.</p> <p>Puede emplearse eficazmente para hacer ciclos en <i>Spiral</i> moviéndonos rápidamente sin detenerse en demasiados detalles.</p>	
Ejemplos	<p>El sentido de los bucles en C++ puede ser apreciado en una secuencia de ejercicios en los que las condiciones de inicialización, pruebas e incremento puedan ser alteradas.</p> <p>Puede comprobarse cuando finaliza un bucle "while" o cuando no, dentro del cuerpo de un programa.</p>	

Nombre del Patrón:	TEST TUBE	#11
Contraindicaciones		
Referencias		
Herramientas LMS propuestas	<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios (almacén de recursos educativos) • V.4 - Herramientas de calificación en línea • I.4 - Intercambio de archivos 	
Propuesta utilización	<p>El profesor recopilará para la asignatura, software de utilización para que el alumno ensaye libremente. Se le solicitarán trabajos donde el alumno responda preguntas del tipo ¿Qué pasaría si...?. Estos programas pueden residir en repositorio o mostrar enlaces a elementos externos. Una buena fuente para ayudar al alumno en sus reflexiones y preguntas es ofrecer un banco de preguntas que puedan ser consultadas mediante autoevaluaciones o foros, aunque el alumno debe ser suficientemente reflexivo como para preguntarse múltiples preguntas de este tipo.</p>	

Nombre del Patrón:	FILL IN THE BLANKS	#12
Descripción	Normalmente los alumnos aprenden un tema complejo ensamblando pequeñas partes para componer un artefacto más grande. Esto fomenta sus habilidades lectoras y escritoras.	
Problema	Los estudiantes novatos necesitan trabajar en proyectos cada vez de mayor envergadura, pero están poco experimentados y no poseen aún el conocimiento y las habilidades necesarias. Por otra parte, pueden aprender leyendo y practicando. ¿Cómo se puede motivar a los estudiantes para trabajar en grandes artefactos sin que se abrumen?	
Audiencia/Contexto	Este modelo está pensado para cursos de programación donde se intenta que los alumnos avancen rápidamente con material difícil de asimilar. Seguramente es posible adaptar este patrón para diseño. También consigue <i>Read Before Write</i> .	
Fuerzas	Los estudiantes tienen difícil disponer de tiempo para desarrollar programas complejos desde cero. Los programas que puedan construir al principio suelen ser bastante aburridos. Los estudiantes deben ver cómo se encaja el trabajo que hacen en un contexto más amplio. Los estudiantes pueden aprender, leyendo los programas desde el principio, que pueden ser capaces de escribirlos. Sin embargo, no se permitirá una actitud demasiado pasiva en su lectura.	
Solución	Preparar un programa muy bien diseñado o parte de un programa y quitar algunas piezas del código. Dar el resultado a los estudiantes con instrucciones para rellenar las partes faltantes. Las partes faltantes deben especificarse bien. Sería mejor si el resultado es utilizado inmediatamente para que puedan apreciar el efecto de su trabajo.	
Discusión / Consecuencias / Implementación	El diseño general del artefacto debe ser excelente. Las piezas faltantes deben seleccionarse cuidadosamente para que los estudiantes puedan deducir algo sobre las partes que faltan de las piezas suministradas. En algunos casos las partes que faltan pueden describirse cuidadosamente con pre y post condiciones. En otros casos, la especificación puede dejarse a los propios estudiantes. Incluso podría ser creados en clase en un <i>Student Design Sprint</i> . Las piezas faltantes podrían precisar de diferentes habilidades para completarlas. Algunas partes que faltan pueden requerir algo más de conocimiento sintáctico, aunque algunos pueden requerir análisis semántico profundo. Si se utiliza varias veces (como en una <i>Spiral</i>) los usos sucesivos podrían requerir cada vez un análisis más profundo.	
Requisitos	Crear cualquier artefacto bien diseñado que puede ser modificado.	
Patrones relacionados	En <i>Fixer Upper</i> , los estudiantes reparan un artefacto cuidadosamente roto. Aquí completan un artefacto dejado sin acabar expresamente. El artefacto también puede ser <i>Larger than Life</i> , <i>Early Bird</i> o <i>Lay of the Land</i> .	
Ejemplos	Un ejemplo simple es la definición de una clase en un lenguaje como C++ o Java en el que algunos del método principal/de implementación hayan sido omitidos. Algo más complejo sería si no se hubiera incluido la parte de la interfaz. Más simple es abandonar una línea aislada de código en algún lugar, aunque se deben proporcionar los comentarios adecuados para que los estudiantes no tengan que adivinar dónde están estos huecos en blanco. Se puede proporcionar una aplicación que requiera ciertas librerías de código (pila, cola, ...) que deberá ser proporcionadas por el estudiante.	

Nombre del Patrón:	FILL IN THE BLANKS	#12
Contraindicaciones		
Referencias		
Herramientas LMS propuestas	<ul style="list-style-type: none"> • VII.3 - Librerías digitales o repositorios (almacén de recursos educativos) • V.4 - Herramientas de calificación en línea. • I.1 - Foros • I.4 - Intercambio de archivos (documentación de apoyo) 	
Propuesta utilización	<p>Se propone al alumno participar en un gran proyecto con pequeñas aportaciones que le obligue a conocer su comportamiento pero no necesariamente conocer a fondo el proyecto. Este proyecto residirá en un repositorio. Toda la información necesaria debe estar documentada, y un foro puede servir de intercambio entre el profesor y el alumno. El alumno deberá completar las partes del proyecto que intencionadamente han quedado en blanco.</p>	

Nombre del Patrón:	GOLD STAR	#13
Descripción	A los estudiantes se les debe elogiar cuando lo hacen bien.	
Problema	A veces los estudiantes sorprenden en clase o al realizar un trabajo. Normalmente se les elogia de forma privada lo cual está muy bien, pero entonces se pierde la oportunidad de mostrar a otros estudiantes que es lo que más valoramos.	
Audiencia/Contexto	Todos los estudiantes de todos los niveles.	
Fuerzas	Los estudiantes buscan en los profesores un modelo a seguir. Quieren y necesitan su aprobación. La alabanza puede ser una motivación, especialmente cuando viene de alguien que es respetado. Los estudiantes trabajan mejor cuando se sienten bien consigo mismo. A los estudiantes, como a sus profesores, les gusta sentirse apreciados.	
Solución	Cuando los estudiantes hacen algo bien, apreciarán los elogios. Ofrecer un elemento de reconocimiento por el trabajo bien hecho. Esto puede realizarse de forma pública o privada. Puede ser unas simples palabras pronunciadas en privado o puede ser una mención en el expediente académico del estudiante. Las fuerzas en juego aquí sugieren también que nunca se debe menospreciar al estudiante ni siquiera por un trabajo pobre.	
Discusión / Consecuencias / Implementación	Esto funciona para niños pequeños, tal como sabe todo profesor de escuela primaria. A menudo es inesperado por adultos jóvenes, y esto solo puede explicar algunos casos con eficacia.	
Requisitos		
Patrones relacionados		
Ejemplos	<p>Un “medallero de oro, plata y bronce” con los ganadores cada año de un curso en internet (p.e. http://CSIS.Pace.edu/~Bergin/Compiler/CompilerAward.html)</p> <p>Al cruzarse con un estudiante en el ascensor o en las aulas, se puede mencionar cuánto ha gustado una solución que proporcionaban en un ejercicio anterior (si en efecto lo hizo). Cosas sencillas, como esta pueden (y hacen) cambiar vidas.</p> <p>Dartmouth College tiene una política por la que un profesor puede insertar una mención especial en las actas de cualquier estudiante cuando hacen algo fuera de lo común. Puede considerarse un trabajo especialmente bueno en un proyecto o una contribución especial a una clase o a la trayectoria en el campus. Estas notas son muy apreciadas por los estudiantes.</p> <p>Un ejemplo extravagante ejemplo ocurre cuando un profesor publica los mejores trabajos de los alumnos en una página web del curso.</p> <p>El nombre, por supuesto, proviene de una práctica bastante común en la escuela primaria en la que un profesor pondrá una estrella dorada en fichas bien hechas.</p> <p>Algunos profesores utilizan marcas de tinta especial con anotaciones especiales para sus estudiantes. Ofrecer de manera pública algunas estrellas de oro en cada curso (disponible en cualquier papelería), principalmente a hacer preguntas relevantes.</p>	

Nombre del Patrón:	GOLD STAR	#13
Contraindicaciones	Si va a hacer esto, no olvide ser coherente.	
Referencias		
Herramientas LMS propuestas	<ul style="list-style-type: none"> • V.4 - Herramientas de calificación en línea • I.1 - Foros • III.1 - Grupos de trabajo • III.4 - Perfil del estudiante (orla) 	
Propuesta utilización	Se debe premiar a los alumnos que realizan los mejores trabajos con reconocimientos en foros /blogs /orlas, publicando los mejores trabajos para que el resto de alumnos reconozcan un buen aporte. Incluso reagrupar los alumnos en función de estos aportes para trabajar en grupos selectos. También se puede dar publicidad de estos reconocimientos con otras herramientas como noticias, novedades, avisos, etc.	

Nombre del Patrón:	GRADE IT AGAIN, SAM	#14
Descripción	Proporcionar un marco en el que los alumnos puedan equivocarse y aprender de sus propios errores, permitiéndolos volver a presentar reiteradamente a las evaluaciones para obtener mejor puntuación.	
Problema	Los estudiantes están en clase para aprender. Cometan errores. A veces no corren riesgos porque temen ser suspendidos. Se desea ayudar a los estudiantes a que aprendan, y cuando para ello trabajan duro, se les debe recompensar por su éxito, especialmente si ha sido difícil de obtener.	
Audiencia/Contexto	Este patrón puede utilizarse en cada curso. Este patrón ayuda a reducir el riesgo (<i>Reduce Risk</i>). La única excepción es cuando el curso se evalúa por la preparación de un gran trabajo, pues la valoración final dependerá del estado de este trabajo. Para esa situación ver <i>Fair Project Grading</i> .	
Fuerzas	Se supone que un colegio o universidad, es un lugar seguro en el que aprender. Esto implica cometer errores. Los alumnos no llegan a un curso para demostrar que no es necesario realizar el curso. Todos cometen errores, los novatos más de nadie. Los alumnos deben tener un entorno en el que pueden aprender de sus errores. Los profesores están ocupados y tienen un montón de cosas que hacer además de enseñar.	
Solución	Cuando se activa un ejercicio o examen, permitir a los estudiantes la reevaluación siempre que presente cambios en su ejercicio o trabajo, para obtener mejor nota o subir de nivel. La calificación podrá ser mayor que la anterior, pero se puede contemplar una penalización de modo que la “puntuación perfecta” no sea posible en caso de reevaluación. Se puede poner un límite en el número de reevaluaciones. En una clase de gran tamaño (más de 30 alumnos, sin profesor auxiliar) quizá deba limitarse a 1 el número de reevaluaciones. En una clase menor puede no ser necesario limitar el número de reevaluaciones. Una variación es permitir al estudiante reenviar un trabajo hasta una fecha determinada.	
Discusión / Consecuencias / Implementación	<p>Algunos estudiantes se aprovechan del sistema y se benefician reevaluándose porque no tienen nada que perder. Los profesores pueden contar con un historial de cada estudiante a fin de valorar el esfuerzo y si se han tenido en cuenta sus propuestas de cambio en los trabajos. También ayuda si los alumnos utilizan control de cambios (cambio de barras).</p> <p>A veces la posibilidad de mejora es pequeña y el alumno debe invertir su tiempo en otras tareas.</p> <p>A veces el profesor no tiene tiempo para las reevaluaciones.</p> <p>El propósito de este patrón es permitir al estudiante dedicar un esfuerzo adicional sobre los materiales con los que tienen dificultad especial.</p> <p>Es importante tener en cuenta que el nombre es un juego en la película Casablanca, en la que Rick (Humphrey Bogart) dice repetidamente su pianista "Play it again, Sam". Woody Allen también tiene una película con este nombre.</p>	
Requisitos	Se precisa de tiempo y mano de obra intensiva.	
Patrones relacionados		
Ejemplos		

Nombre del Patrón:	GRADE IT AGAIN, SAM	#14
Contraindicaciones	<p>Si los estudiantes son descuidados puede presentar un trabajo pobre la primera vez, pensando que no hay ningún riesgo. Utilizar una penalización de 10% en la nota para evitar este comportamiento. Si los estudiantes son excesivamente conservadores entonces pueden tardar demasiado tiempo en enviar nuevas ediciones y corren el riesgo descuidar otros trabajos. Para evitar esto, por ejemplo, puede no darse el último tres por ciento en trabajos reenviados.</p>	
Referencias		
Herramientas LMS propuestas	<ul style="list-style-type: none"> • III.2 - Autovaloraciones. • V.4 - Herramientas de calificación en línea • I.1 – Foro 	
Propuesta utilización	<p>Se le ofrecerá un banco de preguntas para realizar ejercicios de autoevaluación. Otra opción es posibilitar al alumno presentarse distintas veces al examen para subir nota. Un foro entre alumnos puede ser útil para aclarar dudas. El profesor debe adoptar una posición de segunda línea. El alumno debe conocer a priori las reglas de reevaluación y calificación dependientes del trabajo y número de intentos.</p>	

5. Validación

Pensadores como René Descartes propusieron métodos para alcanzar un conocimiento considerado válido. Lo que hoy conocemos como *método científico*, es producto de siglos de desarrollo del pensamiento y de la experimentación. Grandes personajes, entre ellos Francis Bacon, Isaac Newton, Galileo Galilei, Francesco Redi, Charles Darwin, Louis Pasteur y tantos otros, fueron sentando las bases de un método basado en dos principios básicos: la objetividad y la racionalidad [96].

Estos científicos demostraron que a través de la observación y la experimentación, (es decir, el someter a validación empírica la relación entre los hechos) era posible llegar a explicar y predecir los fenómenos de la naturaleza. El conocimiento científico llegaría, poco a poco, a desplazar las nociones derivadas de las creencias religiosas y las disciplinas precientíficas.

Siguiendo este proceder, hemos considerado necesario someter a la valoración de distintos profesionales de la educación con sólidos conocimientos en el ambiente del e-learning nuestras propuestas tanto en la utilidad de los patrones descritos por Bergin, como las herramientas utilizadas para aplicar estos patrones así como una pequeña guía de cómo combinar estas herramientas.

La forma de hacer llegar esta propuesta y conocer la opinión de estos profesionales ha sido la de encuesta online, básicamente por los siguientes motivos:

1. Esta captura de información se ha realizado en poco tiempo para la publicación en este trabajo.
2. La amplia dispersión geográfica en la que se encontraban las personas encuestadas.

La encuesta consta de dos apartados: en la primero se solicitan datos sociológicos (sexo, vinculación a la universidad, experiencia y área docente) y en la segunda se solicita una valoración de para cada patrón y de las herramientas propuestas. La valoración varía del 1 al 5, siendo 5 la más alta y 1 la más baja, según se muestra en la Figura 3. Además, la encuesta insta a proponer alguna herramienta nueva y como aplicarla. Finalmente, la encuesta solicita una valoración global de la encuesta donde pudiera manifestar aspectos no contemplados en la propia encuesta.

Ayuda

1. Early Bird
2. Spiral
3. Consistent Metaphor
4. Toy Box
5. Tool Box
6. Lay of the Land
7. Fixer Upper
8. Larger Than Life
9. Student Design Sprint
10. Mistake
11. Test Tube
12. Fill in the Blanks
13. Gold Star
14. Grade it Again Sam

Ayuda Herramientas LMS

***Obligatorio**

Patrón 2 - Spiral

PLANTEAMIENTO
Dividir el temario en fragmentos que se irán introduciendo poco a poco para resolver un problema. Se volverá sobre el mismo tema pero con más profundidad.

PROPUESTA
Se dispondrá de un glosario con los términos comunes donde el alumno pueda ampliar información. Si se quiere que todos los alumnos participen en la elaboración de este glosario puede utilizarse wikis. Realizaremos documentos breves con continuos hipervínculos entre ellos, y por esto es necesario definir un guión del curso de forma que el alumno no se disperse.

HERRAMIENTAS EMPLEADAS

- I.9 - Wikis (Glosario)
- I.4 - Intercambio de archivos (programa y cronograma)

Por favor, conteste a las siguientes preguntas *

	totalmente de acuerdo	de acuerdo	indiferente	en desacuerdo	totalmente en desacuerdo
2.1 - Cree útil este patrón para su docencia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.2 - Le parecen adecuadas las herramientas propuestas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2.3 ¿ Desea proponer otro tipo de herramienta ?

<Ninguna>

2.4 - Indique cómo la aplicaría.

Figura 3. Cuestionario de valoración.

La encuesta se extendió a profesionales de la Universidad Complutense de Madrid, Universidad Nacional de La Plata en Buenos Aires, Universidad Politécnica de Valencia y Universidad Rey Juan Carlos de Madrid. Concretamente, en la Universidad Complutense de Madrid fueron encuestados los coordinadores del Campus Virtual de la UCM en los centros, es decir, docentes de los distintos centros encargados de impulsar el uso del Campus Virtual dentro de la propia universidad [81].

Cuarenta y dos personas respondieron a la encuesta. El perfil del encuestado responde a un funcionario, hombre con más de 10 años de experiencia perteneciente al área científica e ingenierías, tal como se muestra en la Tabla 4.

Variables Sociológicas

Sexo	Frecuencia	Porcentaje
Hombre	27	64,3
Mujer	15	35,7
Total	42	100,0
Vinculación	Frecuencia	Porcentaje
Colaborador/Becario	5	11,9
Contratado	11	26,2
Funcionario	16	38,1
Otros	10	23,8
Área	Frecuencia	Porcentaje
Biosanitaria	7	16,7
Científica e Ingenierías	22	52,4
Ciencias Sociales y Humanitarias	13	31,0
Experiencia	Frecuencia	Porcentaje
Menos de 3 años	4	9,5
De 3 a 5 años	5	11,9
De 5 a 10 años	8	19,0
De 10 a 20 años	13	31,0
Más de 20 años	12	28,6

Tabla 4. Variables sociológicas

Como resultado general de la encuesta, se puede afirmar que existe una valoración positiva del uso de patrones (64,12%) en su práctica docente diaria, si bien no se ha observado una inclinación por ningún patrón concreto (Figura 4).

Patrón	Valoración media patrón ¹
1 Early bird	3,71
2 Spiral	3,62
3 Consistent metaphor	3,36
4 Toy box	4,17
5 Tool box	3,60
6 Lay of the land	3,95
7 Fixer upper	3,71
8 Larger than life	3,93
9 Student design sprint	3,55
10 Mistake	3,10
11 Test tube	3,48
12 Fill in the blanks	3,57
13 Gold star	3,93
14 Grade it again, Sam	3,86
Media	3,68

¹ Valoración de 1 a 5

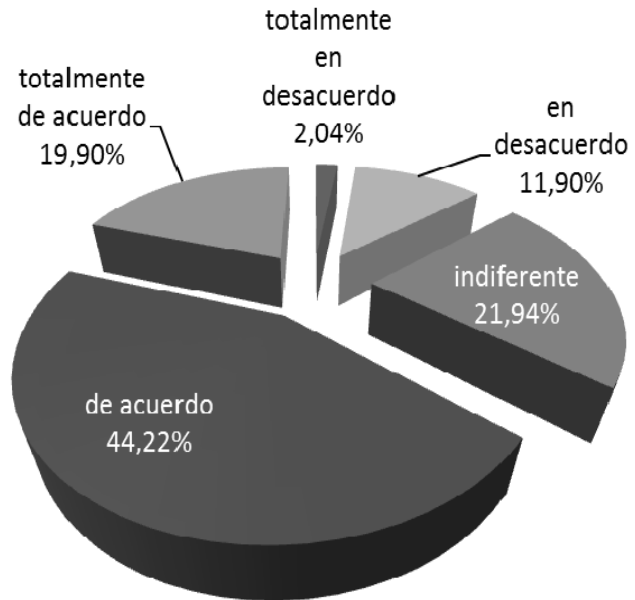
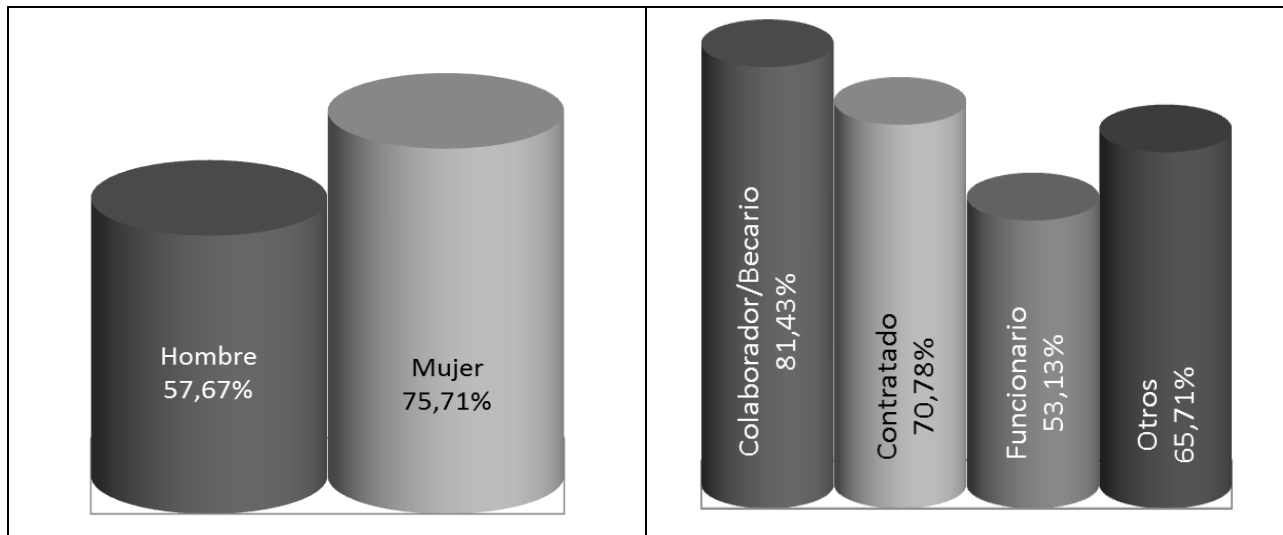


Figura 4. Valoración global de los patrones de Bergin

Considerando los 4 ejes sociales (sexo, vinculación, experiencia y área) que hemos solicitado a los encuestados, se observa: (i) que las mujeres valoran más positivamente la utilización de los patrones que los hombres; (ii) que los funcionarios son más reticentes al uso de patrones que otros colectivos; (iii) que a más años de experiencia menos interés por la aplicación de estos patrones; y (iv) que la valoración positiva de estos patrones es prácticamente independiente del área docente (Figura 5).



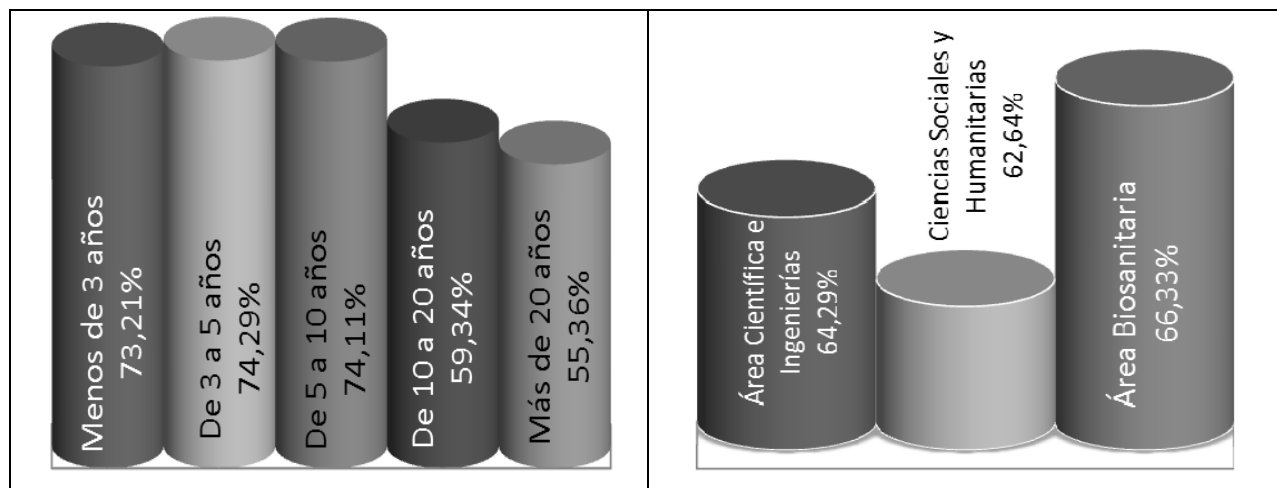


Figura 5. Valoración positiva de patrones de Bergin por colectivo.

Para obtener las diferencias entre estas variables sociales y la valoración de los patrones y herramientas propuestas, hemos cruzado cada una de las variables con cada patrón con el programa de tratamientos estadísticos SPSS empleando técnicas multivariantes y árboles de segmentación tal como se muestra en la Figura 6.

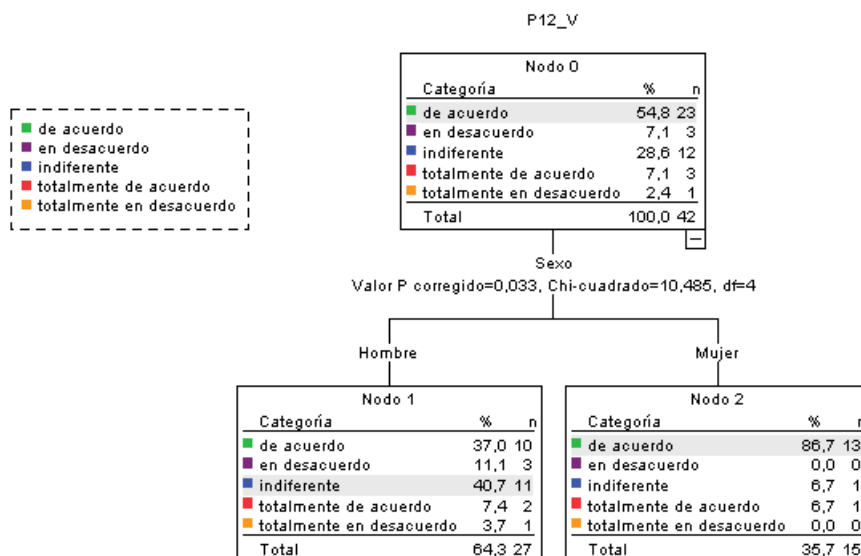


Figura 6. Cruce de patrones con variables sociales utilizando SPSS.

Ya dentro de la valoración de los patrones, en la Figura 6 hemos podido constatar una diferencia importante respecto al sexo con uno de los patrones (*Fill in the blanks*) donde un 93,4% de mujeres se mostraban de acuerdo o totalmente de acuerdo, frente a un 44,4% de hombres.

Con respecto a la vinculación, se encuentran diferencias apreciables en el patrón *Fixer Upper* donde el colectivo funcionario es el único que se muestra contrario a la aplicación de este patrón en un 38% de casos.

El área al que aplican la docencia los encuestados, no arroja ninguna diferencia destacable respecto a la apreciación de los patrones. Tampoco se muestran diferencias destacables respecto a la experiencia docente debido, en gran parte, a que más de la mitad de los encuestados tienen más de 10 años de experiencia. Así en el patrón *Early Bird* la aceptación va disminuyendo desde el 100% de docentes con menos de 5 años de experiencia a un 41,7% de docentes con más de 20 años de docencia,

Con respecto a la valoración de las herramientas propuestas para la aplicación de los patrones y su forma de combinarlas, podemos afirmar que los porcentajes de aceptación global y por colectivos permanecen invariantes con respecto a la valoración de patrones y, en general, se consideran adecuados para conseguir la aplicabilidad del patrón correspondiente (Figura 7).

Patrón	Valoración media herramientas ¹
1 Early bird	3,83
2 Spiral	3,62
3 Consistent metaphor	3,57
4 Toy box	4,07
5 Tool box	3,64
6 Lay of the land	3,76
7 Fixer upper	3,71
8 Larger than life	3,86
9 Student design sprint	3,74
10 Mistake	3,50
11 Test tube	3,52
12 Fill in the blanks	3,71
13 Gold star	3,98
14 Grade it again, Sam	3,93
Media	3,75

¹ Valoración de 1 a 5

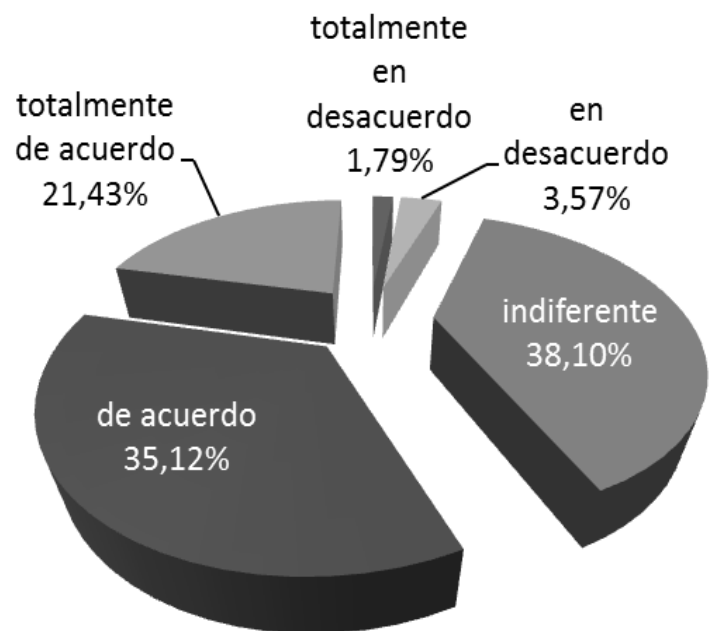


Figura 7. Valoración global de herramientas para aplicar los patrones de Bergin

De nuevo las cuatro variables sociales utilizadas no muestran gran diferencia en cuanto a la valoración de las herramientas propuestas (Figura 8), descubriéndose una mayor aceptación en mujeres y en colectivos con menos años de experiencia docente. Representativos es que todos los

colectivos valoran las herramientas propuestas por encima incluso de la valoración de los patrones.

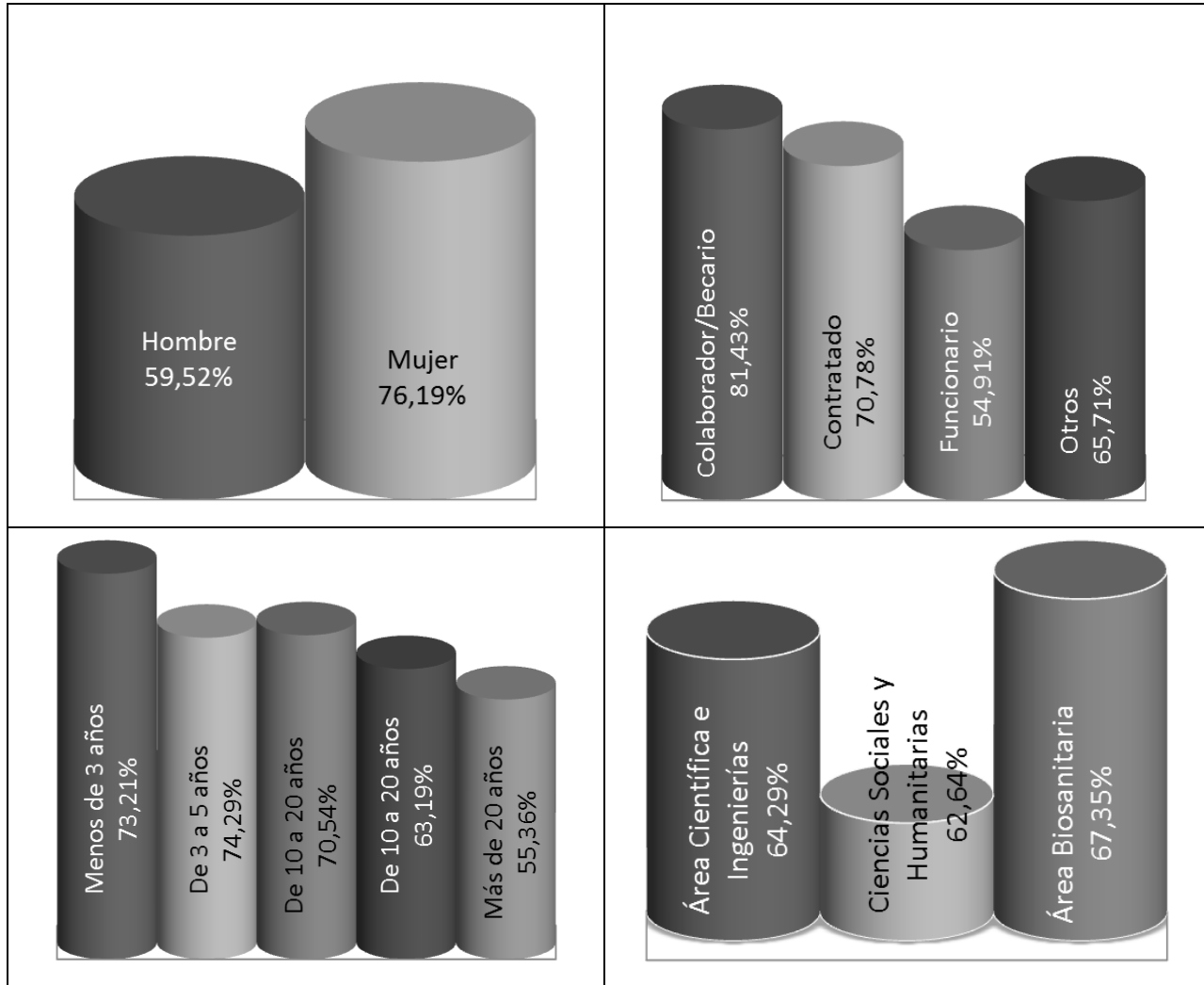


Figura 8 Valoración positiva de herramientas para aplicar los patrones de Bergin por colectivo.

En cuanto a la valoración por patrones de las herramientas propuestas, hemos podido constatar que no existen diferencias importantes en cuanto al sexo (la diferencia más notable surge en el patrón *Fill in the blanks* donde un 86,7% de mujeres se declaran de acuerdo o totalmente de acuerdo frente a un 55,5% de hombres). El tipo de relación con la universidad es un factor importante en la valoración de las herramientas propuestas para el patrón *Fixer Upper* donde los Contratados o Becarios están de acuerdo en un 100% frente a un 50% de los funcionarios (en consonancia con la valoración del propio patrón), El área no ha sido una

variable decisiva respecto a la valoración de las herramientas propuestas. Con respecto a los años de docencia, no se aprecia la diferencia que observábamos con respecto al patrón *Early Bird*, pues la valoración positiva es prácticamente uniforme con independencia de los años de experiencia docente.

Si observamos la valoración de los patrones por parte de los encuestados en la Figura 9, podemos apreciar que dos patrones se encuentran por debajo del 50% de aceptación (*Mistake* y *Consistent metaphor*), pero ninguno de ellos supera el 50% de rechazo, es decir, no lo aplicarían pero no consideran que no pueda ser aplicado en otro tipo de asignaturas.

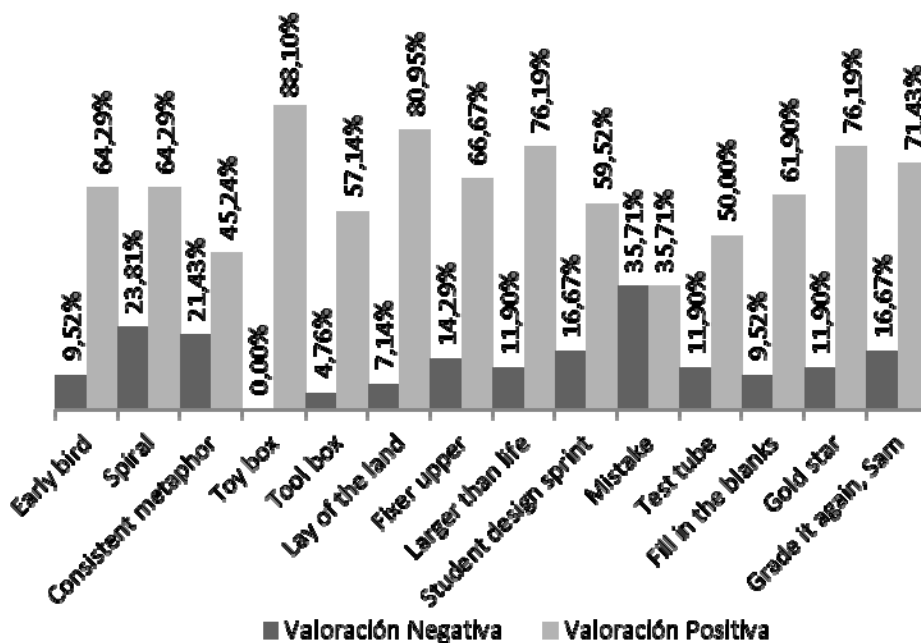


Figura 9 Valoración individual de los patrones.

A pesar de la baja valoración de los patrones citados, en la Figura 10 podemos observar la satisfacción con las herramientas propuestas, es decir, si aplicarían estos patrones en su actividad docente, seguramente seguirían las recomendaciones propuestas en cuanto a herramientas LMS y forma de combinarlas.

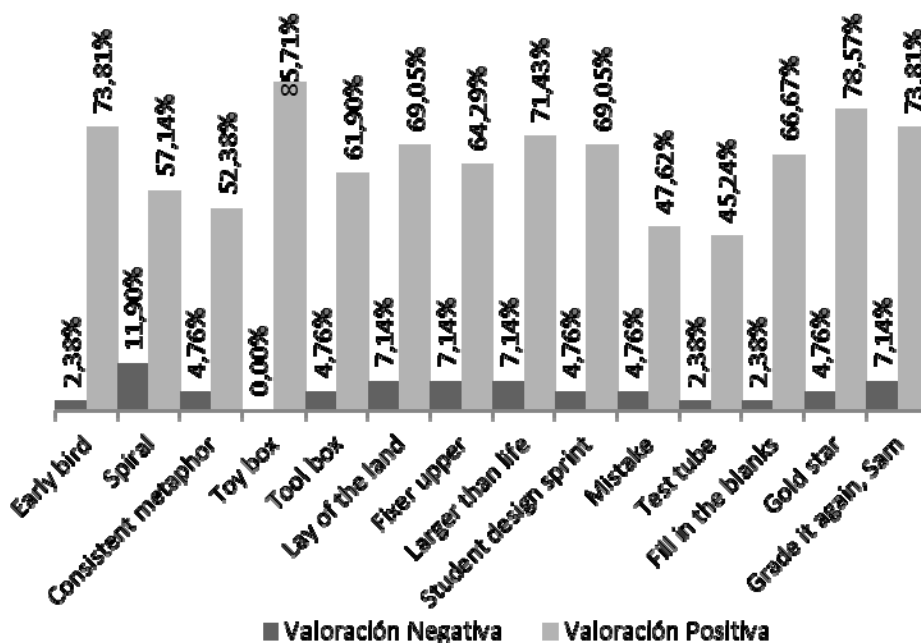


Figura 10 Valoración individual de herramientas aplicables a cada patrón

En la Tabla 5 podemos comprobar cuáles han sido las herramientas más propuestas por los encuestados para poner en práctica los patrones. Las herramientas más demandadas en las contestaciones a la encuesta proponen formación de grupos, servicios de presentación multimedia y creación de foros como herramientas adicionales para la aplicación de los patrones puesto que de esta forma el estudiante se involucra y participa con mayor ahínco en el proceso de aprendizaje. Las estrategias de aprendizaje colaborativo en el aula (y fuera de ella) constituyen un elemento esencial de todo curso universitario. Al trabajar en grupo, los estudiantes ejercitan habilidades interpersonales, de tolerancia, de cooperación, de manejo de conflictos y muchas otras que les serán de gran utilidad en su vida cotidiana y en su futuro trabajo profesional. Además, al interactuar con otros compañeros, se obtiene retroalimentación, se discuten puntos de vista y se complementan y amplían los conceptos. El trabajar en grupo obliga a los estudiantes a estudiar y prepararse, ya que el contacto con la materia del curso no tiene lugar por medio del profesor, de una manera unidireccional (y por ende, pasiva por parte del alumno), sino de ellos mismos. Esto iría en línea con las tesis socioconstructivistas. También son demandados servicios de presentación multimedia lo que demuestra que los profesores consultados desean utilizar este tipo de servicios (pizarra compartida, videoconferencia, etc.) para desarrollar su estrategia docente dentro y fuera de las aulas.

Herramientas LMS	Nº Propuestas
I - Herramientas orientadas al aprendizaje	67
I.1 - Foros.	28
I.2 - Buscador de foros.	1
I.3 - e-portafolio	3
I.4 - Intercambio de archivos	1
I.5 - Herramientas de comunicación síncrona	6
I.6 - Herramientas de comunicación asíncrona	6
I.7 - Servicios de presentación multimedia	9
I.8 - Diarios / Notas en línea	4
I.9 - Wikis	7
I.10 - Encuestas	2
II - Herramientas orientadas a la productividad	16
II.1 - Anotaciones personales o favoritos.	3
II.2 - Calendario y revisión del progreso	4
II.3 - Buscador de cursos	2
II.4 - Mecanismos de sincronización y trabajo fuera de línea	1
II.5 - Control de publicación, páginas caducadas y enlaces rotos	2
II.6 - Noticias del lugar.	3
II.7 - Envío de actualización de páginas, mensajes a foros y envío	1
II.8 - Soporte a la sindicación de contenidos	0
III - Herramientas para la implicación de los estudiantes	19
III.1 - Grupos de trabajo	9
III.2 - Autovaloraciones	6
III.3 - Rincón del estudiante	0
III.4 - Perfil del estudiante	4
IV - Herramientas de soporte	2
IV.1 - Autenticación de usuarios	1
IV.2 - Asignación de privilegios en función del rol del usuario	0
IV.3 - Registro de estudiantes	0
IV.4 - Auditoría	1
V - Herramientas destinadas a la publicación de cursos y contenidos	10
V.1 - Test y resultados automatizados	4
V.2 - Administración del curso	0
V.3 - Apoyo al creador de cursos	1
V.4 - Herramientas de calificación en línea	1
V.5 - Seguimiento del estudiante	4
VI - Herramientas para el diseño de planes de estudio	2
VI.1 - Conformidad con la accesibilidad	1
VI.2 - Reutilización y compartición de contenidos	0
VI.3 - Plantillas de cursos	0
VI.4 - Administración del currículum	0
VI.5 - Personalización del entorno (look & feel).	0
VI.6 - Herramientas para el diseño de la educación	1
VI.7 - Conformidad con el diseño de la educación	0
VII - Sistemas para la gestión del conocimiento en el ámbito educativo	6
VII.1 - Sistemas integrales de conocimiento	1
VII.2 - Sistemas mediadores de información	0
VII.3 - Librerías digitales o repositorios	4
VII.4 - Sistemas basados en ontologías	1
VII.5 - Sistemas basados en folksonomías	0

Tabla 5. Herramientas propuestas por los encuestados.

Para valorar las herramientas propuestas para cada patrón, hemos estimado tener en cuenta aquellas que son “ampliamente” propuestas por los encuestados. En concreto, y para esta primera fase, hemos aceptado aquellas propuestas por más de un 10 por ciento de los encuestados, es decir, por 5 ó más profesores. Y de esta selección obtenemos las siguientes propuestas mostradas en la Tabla 6.

Patrones / Herramientas Propuestas	Nº Propuestas
Early Bird	
I.1 - Foros	8
I.7 – Servicios de presentación multimedia	4
Spiral	
I.1 - Foros	5
Consistent metaphor	
Toy box	
Tool box	
Lay of the land	
Fixer upper	
Larger than life	
Student design sprint	
Mistake	
Test tube	
I.1 – Foros	5
Fill in the blanks	
Gold star	
Grade it again, Sam	

Tabla 6. Valoración de herramientas propuestas por patrón.

De esta manera, debemos incluir estas herramientas en los patrones descritos así como las propuestas redactadas para aplicarlas dentro de cada patrón.

6. Conclusiones y trabajo futuro

Este trabajo se ha centrado en mostrar la utilidad de aplicar patrones en el ámbito docente y, más particularmente, en el ámbito del e-learning en cualquiera de sus facetas (virtual o mixta). En esta línea, hemos mostrado las ventajas de aplicar patrones, pero también sus riesgos y limitaciones. Se ha realizado un seguimiento de los principales patrones pedagógicos mostrando la predilección por los elaborados por Bergin dentro del Proyecto de Patrones Pedagógicos (PPP). Para poder aplicar estos patrones dentro de los Campus Virtuales de cada institución, se ha presentado una categorización de las herramientas que componen un LMS genérico y se ha propuesto la utilización de un grupo diferente de estas herramientas para cada uno de los patrones, junto con una guía para facilitar la aplicación de estas herramientas. Una vez argumentada la propuesta, se ha sometido a valoración por un colectivo de expertos de distintas instituciones universitarias y hemos mostrado la valoración detallada de estas encuestas.

Como principal conclusión, debe señalarse la alta aceptación tanto de la aplicabilidad de los patrones como de la propuesta de herramientas para implantar cada patrón. Es importante destacar el interés de los encuestados por la aplicabilidad de estos patrones, pues todos han propuesto alguna herramienta para mejorar este estudio. Es por ello, por lo que se pretende extender este trabajo mediante la celebración de talleres y experimentando con docentes que deseen ensayar en sus cursos.

Desde el principio se optó, solo, por solicitar cuatro variables al encuestado con el fin de no ahuyentarlo, sin embargo, otros parámetros como el curso de la asignatura, el carácter o el número de créditos hubieran podido ser de interés para conocer mejor donde aplicar estos patrones. Igualmente, la encuesta sólo contemplaba cuatro áreas docentes básicas para facilitar la identificación del encuestado, pero quizá debiera haberse desmenuzado en áreas más específicas para conocer en que estudios o titulaciones existe mayor aceptación y/o comprensión para el uso de patrones.

Un problema con el planteamiento realizado en la elaboración de la encuesta es que se ha solicitado por cada patrón una única herramienta, lo que impide que el encuestado proponga varias herramientas e indique como combinarlas. Por ello, aun recordando que sólo se deben proponer herramientas básicas, se debería ampliar el número de herramientas propuestas. Por

el contrario, si sólo aceptamos añadir herramientas, la complejidad en la aplicación de patrones irá en aumento. Por tanto, sería interesante proponer al encuestado que herramienta o herramientas no considera interesante y por qué.

Ya hemos comentado lo larga que ha resultado la encuesta (entre 20 y 30 minutos) a pesar de la poca información que se proporcionaba para cada patrón. Esto ha podido originar fatiga en el encuestado, por lo que se propone realizar próximas encuestas sobre patrones elegidos de manera aleatoria, de forma que el encuestado pueda conocer toda la información completa sobre cada patrón. Así, podremos evaluar, no solo la utilidad y aplicabilidad de cada patrón, sino también la correcta documentación ofrecida. De esta forma, podríamos profundizar en el Lenguaje de Patrones necesario para catalogar e identificar esos patrones.

Consideramos que la elección de los patrones de Bergin ha sido de gran acierto, pues constituyen un número de buenas prácticas reducidas, documentadas en un lenguaje claro y aceptadas por los docentes encuestados.

Es cierto que los LMS más populares como Sakai, Moodle o Blackboard evolucionan rápidamente y que constantemente surgen aplicaciones de terceros que se pueden integrar para facilitar nuevas funcionalidades o enlazar con servicios alternativos. A pesar de todo ello, consideramos que la categorización presentada para clasificar las herramientas de un LMS genérico mejora la estabilidad de las herramientas propuestas.

En lo referente a las herramientas de los LMSs, las orientadas al aprendizaje han sido las más útiles a la hora de implantar los patrones de Bergin, en particular foros y mecanismos de intercambio de archivos. Dentro del grupo de herramientas pensadas para la implicación de estudiantes, los grupos de trabajo han sido las herramientas más valoradas. En cuanto a las herramientas de publicación de cursos y contenidos, las herramientas de calificación en línea han sido las más utilizadas. Con respecto a las herramientas de gestión del conocimiento, las librerías digitales o repositorios se han mostrado como las más útiles. Por el contrario, las herramientas orientadas a la productividad, las de soporte y las utilizadas para el diseño de planes de estudio no han sido especialmente útiles en la implantación de los patrones de Bergin.

En cuanto al trabajo futuro, nuestros esfuerzos están encaminados en facilitar la creación de espacios docentes virtuales evitando que el profesor sea un experto en el LMS específico para aplicar los patrones pedagógicos de su interés.

De esta forma estamos interesados en analizar más patrones pedagógicos por si pudieran ser del interés de los docentes.

Otra línea de investigación busca traducir los patrones pedagógicos, de carácter general, en patrones e-learning, específicos de la docencia electrónica. En este sentido, resultaría muy interesante el proponer soporte en términos de las herramientas de LMSs genéricos analizadas en este trabajo a los patrones e-learning. De esta forma, podríamos traducir de patrones pedagógicos a patrones e-learning, y de estos a herramientas de LMS. Dicha traducción debería arrojar resultados similares a los obtenidos en este trabajo.

También estamos analizando el uso de técnicas *Model-Driven Architecture* (MDA) para traducir de necesidades de los docentes a configuraciones de plataformas de e-learning. Así, a través del camino *patrones pedagógicos-patrones e-learning-herramientas LMSs*, o bien a través del camino *patrones pedagógicos-herramientas LMSs* podríamos configurar plataformas e-learning concretas en función de las necesidades docentes de los usuarios.

Todo este trabajo cobra especial sentido y relevancia en el contexto de campus virtuales multiplataforma, como los considerados en el proyecto AACV.

Bibliografía

- [1]. **Guri-Rosenblit, S.** Eight Paradoxes in the Implementation Process of E-learning in Higher Education. 2005, Vol. 18, págs. 5-29.
- [2]. *CORDRA: Technical Introduction and Overview.* . 2004, Learning Systems Architecture Lab: Carnegie Mellon LSAL.
- [3]. Instructional Technology Council. *ITC.* [En línea] 2009 Distance Education Survey Results, 2010. [Citado el: 19 de julio de 2011.]
<http://www.itcnetwork.org/attachments/article/66/ITCAnnualSurveyMarch2010Final.pdf>.
- [4]. **Leidner, D. y Jarvenpaa, S.** The use of information technology to enhance management school education: a theoretical view. *MIS Quarterly.* 1995, Vol. 19, págs. 265-291.
- [5]. **Neville, K., Heavin, C. y Walsh, E.** A Case in Customizing E-Learning: Journal of Information Technology. 2005, Vol. 20, págs. 117-129.
- [6]. **Yábar, J. y otros.** The UAB Virtual Campus: An Essential Platform for a European Higher Education Environment. *Journal of Cases om Information Technology.* 2007, Vol. 2, 9, págs. 37-48.
- [7]. Evolución de las TICs en el Sistema Universitario Español 2006-2010. [En línea] 2010. [Citado el: 18 de julio de 2011.]
<http://www.crue.org/export/sites/Crue/Publicaciones/Documentos/Universitic/UNIVERSITIC2010.pdf>.
- [8]. **Bartolomé, A.** Blended Learning. Conceptos básicos. *Pixel-Bit.* [En línea] 2004. [Citado el: 20 de febrero de 2010.]
http://www.lmi.ub.es/personal/bartolome/articuloshtml/04_blended_learning/documentacion/1_bartolome.pdf.
- [9]. **Pietronilla-Penna, M. y Stara, V. J.** *Journal of e-Learning and Knowledge Society* — Vol. 3, n. 2, pp. 127-135. [En línea] 2007. [Citado el: 17 de febrero de 2010.] http://je-lks.maieutiche.economia.unitn.it/index.php/Je-LKS_EN/article/viewFile/254/236.
- [10]. **Cebrián, M.** *Innovar con tecnologías aplicadas a la docencia universitaria.* Madrid : Nancea, 2003. págs. 21-36.

- [11]. **Cabero, J. y Llorente, M.** Las plataformas virtuales en el ámbito de la teleformación. *Alternativas*. [En línea] 2005. [Citado el: 18 de julio de 2011.] <http://www.unicen.edu.ar/b/publicaciones/alternativas/>.
- [12]. The Sakai Project. [En línea] [Citado el: 30 de mayo de 2010.] <http://sakaiproject.org>.
- [13]. Module Object-Oriented Dynamic Learning Environment (Moodle). [En línea] [Citado el: 30 de mayo de 2010.] <http://moodle.org/>.
- [14]. Blackboard learn. [En línea] [Citado el: 30 de 05 de 2010.] <http://www.blackboard.com/Solutions-by-Market/Higher-Education/Blackboard-Learn-for-Higher-Ed/Overview.aspx>.
- [15]. **Alexander, C., Ishikawa, S. y Silverstein, M.** *A pattern language: Towns, buildings, construction*. Oxford : Oxford University Press, 1977.
- [16]. **Gamma, E. y otros.** *Design Patterns : Elements of Reusable Object-Oriented Software*. : Addison-Wesley, 1995.
- [17]. **Rodríguez-Jiménez, J. M.** Patrones pedagógicos en educación virtual. *RED. Revista de Educación a Distancia. Número monográfico X*. [En línea] 2009. [Citado el: 17 de febrero de 2010.] <http://www.um.es/ead/red/M10>.
- [18]. **Erl, T.** *SOA Design Patterns*. Prentice Hall PTR, 2009. ISBN 0136135161.
- [19]. Object Management Group, Inc. *MDA Guide Version 1.0.1*. [En línea] 2003. [Citado el: 19 de julio de 2011.] <http://www.omg.org/cgi-bin/doc?omg/03-06-01>.
- [20]. **Garrison, D. R.** Three generations of technological innovation in distance education. *Distance Education*, 1985, Vol. 16, págs. 235-241.
- [21]. **Garrison, D. R.** *Understanding distance education*. Londres : Routledge, 1989.
- [22]. **Sauvé, L.** Origini e sviluppo dell'instruzione a distanza. *Instruzione a Distanza*. 1992, Vol. IV, 2.
- [23]. **García Aretio, L.** Historia de la educación a distancia. *RIED – Revista Iberoamericana de Educación a Distancia*. [En línea] 1999. [Citado el: 21 de junio de 2011.] <http://www.utpl.edu.ec/ried/images/pdfs/vol2-1/historia.pdf>. ISSN: 1138-2783.
- [24]. **Skinner, B. F.** Proceedings of the Royal Society of London. *Series B, Biological Science*. Julio de 1965, Vol. 162, 989, págs. 427-443.
- [25]. **Area, M.** Introducción a la Tecnología Educativa. [En línea] 2009. [Citado el: 21 de junio de 2011.]

- http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Libros/TEdu_Las_TICs_en_la_Educación_MArea.pdf.
- [26]. **Shklar, L. y Rosen, R.** *Web Application Architecture: Principles, Protocols and Practices*. Wiley, 2009.
- [27]. **Rengarajan, R.** LCMS and LMS: Taking advantage of tight integration. *Click 2 Learn*. [En línea] 2001. [Citado el: 20 de julio de 2011.] http://www.e-learn.cz/soubory/lcms_and_lms.pdf.
- [28]. **García Peñalvo, F. J.** Estado actual de los sistemas e-learning. *Teoría de la Educación: educación y cultura en la sociedad de la información*, vol. 6, n 2. [En línea] 2005. http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm.
- [29]. **Khan, B. H.** *Web-based Training*. Englewood Cliffs : Educational Technology Publications, 2001.
- [30]. **Anthony, D. L.** Patterns for Classroom Education. *Pattern Languages of Program Conference*. [En línea] 1995. [Citado el: 20 de febrero de 2010.] <http://ianchaiwriting.50megs.com/classroom-ed.html>.
- [31]. **Kerievsky, J.** Knowledge Hydrant: A Pattern Language for Study Groups. [En línea] 1999. [Citado el: 17 de febrero de 2010.] <http://www.industriallogic.com/papers/khdraft.pdf>.
- [32]. **Fricke, A. y Voelter, M.** SEMINARS: A Pedagogical Pattern Language about teaching seminars. [En línea] 2000. [Citado el: 12 de octubre de 2010.] <http://www.voelter.de/data/pub/tp/tp.pdf>.
- [33]. **Eckstein, J.** Learning to Teach and Learning to Learn: Running a Course. [En línea] 2000. [Citado el: 17 de febrero de 2010.] <http://www.pedagogicalpatterns.org/examples/LearningAndTeaching.pdf>.
- [34]. **Bergin, J.** Fourteen Pedagogical Patterns. [En línea] 2002. [Citado el: 17 de febrero de 2010.] <http://www.csis.pace.edu/~bergin/PedPat1.3.html>.
- [35]. Generative Learning Objects. [En línea] [Citado el: 30 de mayo de 2010.] <http://www.rlo-cetl.ac.uk/whatwedo/glos/index.php>.
- [36]. **Dodge, B.** WebQuest: A Technique for Internet-based Learning. *Distance educator*. 1995, Vol. 1 (2), págs. 10-13.

- [37]. **Jones, D. y Stewart, S.** The case for patterns in online learning. [En línea] 1999. [Citado el: 20 de febrero de 2010.]
http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/16/c2/ff.pdf.
- [38]. The E-LEN Project. [En línea] 2005. [Citado el: 20 de febrero de 2010.]
<http://www2.tisip.no/ELEN/>.
- [39]. The E-Dilema Project. [En línea] 2002. <http://e-dilema.uhk.cz/default.asp>.
- [40]. The Person-Centered e-Learning Pattern Repository. [En línea] 2006. [Citado el: 20 de febrero de 2010.] <http://elearn.pri.univie.ac.at/patterns/>.
- [41]. **Boneu, J.** Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos. *Contenidos educativos en abierto*. [En línea] 2007. [Citado el: 08 de junio de 2011.] <http://www.uoc.edu/rusc/4/1/dt/esp/boneu.pdf>.
- [42]. **Jiménez-Rodríguez, J. y Rico-Pérez, C.** Cómo desarrollar un portafolio digital: un ejemplo práctico basado en blogs. *XVI Congreso Internacional de ASELE*. [En línea] 2005. [Citado el: 10 de junio de 2011.]
http://cvc.cervantes.es/ensenanza/biblioteca_ele/asele/pdf/16/16_0758.pdf.
- [43]. Wikipedia. *USENET NewsGroup*. [En línea] [Citado el: 10 de mayo de 2011.]
http://en.wikipedia.org/wiki/Usenet_newsgroup.
- [44]. **Von Elek, T.** A Test of Swedish as a second language: an experiment in self-assessment. [ed.] Y.P. Lee et al. *New directions in language testing*. Oxford : Pergamo Institute of English, 1985, págs. 47-57.
- [45]. **Berners-Lee, T., Hendler, J. y Lassila, O.** The Semantic Web. *Scientific American*. [En línea] 17 de mayo de 2001. [Citado el: 21 de junio de 2011.]
<http://www.dblab.ntua.gr/~bikakis/SW.pdf>.
- [46]. **Cáceres Tello, J.** Patrones de diseño: ejemplo de aplicación en los Generative Learning Object. *Revisa de Educación a Distancia (RED)*. [En línea] 2010. [Citado el: 23 de julio de 2011.] <http://www.um.es/ead/red/M10/caceres.pdf>.
- [47]. **Appleton, B.** Patterns and Software: Essential Concepts and Terminology. [En línea] 2000. [Citado el: 17 de febrero de 2010.]
<http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>.
- [48]. **Alexander, C.** *A Timeless Way of Building*. Oxford : Oxford University Press, 1979.

- [49]. **Cristóbal, J., Navarro, A. y Buendía, F.** Caracterización de Catálogos de Patrones de e-learning en Instituciones Universitarias. *Jornadas de Ingeniería del Software y Bases de Datos, Vol. 4, No. 3.* [En línea] 2010. [Citado el: 23 de julio de 2011.] http://www.sistedes.es/TJISBD/Vol-4/No-3/articles/iselear10_submission_6.pdf. ISSN 1988-3455.
- [50]. **Coplien, J. O.** Design Pattern Definition: Software Patterns. *Hillside Group: Home of the Patterns Library.* [En línea] 2003. [Citado el: 24 de julio de 2011.] <http://hillside.net/component/content/article/50-patterns/222-design-pattern-definition>.
- [51]. **Cueva Lovelle, J. M.** *Tecnología de Objetos: Patrones de Diseño.* 2004.
- [52]. **Montero, S. y otros.** *Patrones de diseño aplicados al desarrollo de ODES y entornos educativos basados en TIC.* Intituto de Investigaciones Educativas (ITE). Serie Informes 21.
- [53]. **Rogers, C. R.** *Freedom to learn for the 80's.* [ed.] F. Charles. Columbus : Merrill Publishing Company, 1983.
- [54]. The Pedalogical Patterns Project. [En línea] [Citado el: 04 de agosto de 2011.] <http://www.pedagogicalpatterns.org/>.
- [55]. Hypermedia Design Patterns Repository. [En línea] [Citado el: 04 de agosto de 2011.] <http://c2.com/cgi/wiki?HypermediaDesignPatternsRepository>.
- [56]. **Duyne, D., Landay, J. y Hong, J.** *The Design of Sites: Patterns, Principles and Processes for Crafting a Customer-Centered Web Experience.* Addison-Wesley Professional, 2002. ISBN-13: 978-0201721492.
- [57]. **De Bra, P. y otros.** AHA! Version 2.0: More Adaptive Flexibility for Authors. [ed.] Proc. AACE Elearn. 2002.
- [58]. **De Bra, P. y Calvi, L.** Creating adaptive hyperdocuments for and on the Web. Proc. WebNet' 1997, 1997, págs. 149-155.
- [59]. **De Bra, P. y Calvi, L.** Towards a Generic Adaptive Hypermedia System Hypertext and Hypermedia. Proc. Workshop on Adaptive, 1998, Vols. 5-1 1.
- [60]. **Brusilovsky, P. y Pesin, L.** ISIS-Tutor: An adaptive hypertext learning environment. Proc. JCKBSE, 1994, págs. 83-87.
- [61]. **Brusilovsky, P., Schwarz, E. y Weber, G.** ELM-ART: An intelligent tutoring system on World Wide Web. Proc. ITS, LNCS 1086, 1996, págs. 261-269.

- [62]. **Brusilovsky, P., Eklund, J. y Schwarz, E.** Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, 30(1-7), págs. 291-300.
- [63]. **Bull, S. y otros.** User Modelling in I-Help: What, Why, When and How. Proc. UM. LNAI 2109, 2001, págs. 117-126.
- [64]. **Kobsa, A., Müller, D. y Nill, A.** KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. Proc. UM, 1994, págs. 31-36.
- [65]. **Specht, M. y otros.** Adaptive Learning Environment for Teaching and Learning in WINDS. Proc. Conference on Adaptive Hypermedia and Adaptive Web Based Systems, 2002.
- [66]. **Spech, M. y otros.** Adaptive WWW-Courseware for Statistics. Proc. UM, 1997.
- [67]. **Avgeriou, P. y otros.** Design patterns in adaptative Web-based Educational Systems: An overview. *First International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia*. [En línea] 16-18 de febrero de 2004. [Citado el: 04 de agosto de 2011.] <http://www.cs.rug.nl/~paris/papers/WBE04.pdf>.
- [68]. **Buschmann, F. y otros.** Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley & Sons, 1996.
- [69]. **Schmidt, D. y otros.** Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects. John Wiley & Sons, 2000.
- [70]. **Rossi, G., Garrido, A. y Carvalho, S.** Design Patterns for Object-Oriented Hypermedia Applications. [ed.] J. Vlissides, J. Coplien y N. Kerth. *Pattern Languages of Program Design. 2*. Addison-Wesley, 1996, págs. 177-191.
- [71]. **Jung, C. G.** *El hombre y sus símbolos*. Londres : Aldus, 1964.
- [72]. **Forman, R. T. y Godron, M.** *Ecología del Paisaje*. New York : John Wiley, 1986.
- [73]. **Lorenz, K.** *Sobre la agresión*. Londres : Methuen, 1963.
- [74]. **Crowe, S.** *El patrón de paisaje*. Chichester. : Packhard, 1988.
- [75]. **Thompson, D.** *El crecimiento y la forma*. Cambridge : Cambridge University Press, 1961.
- [76]. **Moura, C. O. y Derycke, A.** Pedagogical Patterns and Learning Design: When Two Worlds Cooperate. *Workshop UNFOLD/PROLEARN*. [En línea] 22-23 de septiembre de

2005. [Citado el: 08 de agosto de 2011.]
http://dspace.ou.nl/bitstream/1820/474/9/09_DBU_review.pdf.
- [77]. **McAndrew, P., Goodyear, P. y Dalziel, J.** Patterns, designs and activities: unifying descriptions of learning structures. *International Journal of Learning Technology*, 2(2-3). 2006, págs. 216-242.
- [78]. **Skinner, B. F.** *About behaviourism*. New York : Knopf, 1974.
- [79]. **Salgado, E.** *Manual de Docencia Univeristaria: Introducción al constructivismo en la Educación Superior*. Costa Rica : ULACIT, 2006.
- [80]. **Van Dusen, G.** The Virtual Campus: Technology and Reform in Higher Education. *ASHE-ERIC Higher Education Report*. Washington D.C. : The George Washington University, 1997, Vol. 25, 5, pág. 185.
- [81]. **Fernandez-Valmayor, A., Cristóbal, J. y Navarro, A.** El Campus Virtual de la Universidad Complutense de Madrid. *PixelBit - Revista de Medios y Educación*. Nº 32. [En línea] Marzo de 2008. [Citado el: 04 de agosto de 2011.]
<http://www.sav.us.es/pixelbit/pixelbit/articulos/n32/4.pdf>. ISSN 1133-8482.
- [82]. **Graziadei, W. D. y otros.** Building Asynchronous and Synchronous Teaching-Learning Environments: Exploring a Course/Classroom Management System Solution. [En línea] 1997. [Citado el: 16 de agosto de 2011.]
http://horizon.unc.edu/projects/monograph/CD/Technological_Tools/Graziadei.html.
- [83]. Intelligent Tutoring Systems. [En línea] Association for the Advancement of Artificial Intelligence (AAAI), 2000. [Citado el: 18 de agosto de 2011.]
<http://aaai.org/AITopics/IntelligentTutoringSystems>.
- [84]. **Tidwell, J.** Common Ground: A Pattern Language for Human-Computer Interface Design. [En línea] 1999. [Citado el: 15 de agosto de 2011.]
http://www.mit.edu/~jtidwell/ui_patterns_essay.html.
- [85]. **Prendes Espinosa, M.P.** Plataformas de Campus Virtual con herramientas de software libre: Análisis comparativo de la situación actual en las universidades españolas. [En línea] Secretaría de Estado de Universidades e Investigación, 2009. [Citado el: 18 de agosto de 2011.] <http://www.um.es/campusvirtuales/informe.html>.
- [86]. **Fowler, M.** *Analysis Patterns: Reusable Object Models*. Addison-Wesley, 1996.

- [87]. **Eriksson, H. y Penker, M.** *Business Modelling with UML-Business Patterns at work*. New York : John Wiley & Sons, 2000.
- [88]. **Folmer, E., van Welie, M. y Bosh, J.** Bridging patterns: An approach to bridge gaps between SE and HCI. [En línea] 07 de noviembre de 2005. [Citado el: 18 de julio de 2011.] <http://www.cs.rug.nl/search/uploads/Publications/folmer2006bpa.pdf>.
- [89]. **Kerievsky, J.** *Refactoring to Patterns*. Addison-Wesley Professional, 2004.
- [90]. **Beranek, B. y Newman, Inc.** A history of ARPANET: The first decade. [En línea] 01 de abril de 1981. [Citado el: 29 de agosto de 2011.] <http://www.darpa.mil/WorkArea/DownloadAsset.aspx?id=2677>.
- [91]. History of virtual learning environments. [En línea] [Citado el: 28 de agosto de 2011.] <http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W110119H.pdf>.
- [92]. **Burgos, D.** Extensión de la especificación IMS Learning Design desde la Adaptación e Integración de Unidades de Aprendizaje. *Universidad Carlos III de Madrid. Escuela Politécnica Superior*. [En línea] 2008. [Citado el: 01 de septiembre de 2011.] http://e-archivo.uc3m.es/bitstream/10016/5143/1/Thesis_Daniel_Burgos_Solans.pdf.
- [93]. **Biscay, C.E.** Los estándares e-learning. [En línea] [Citado el: 12 de agosto de 2011.] <http://www.palermo.edu/ingenieria/downloads/CyT5/CYT506.pdf>.
- [94]. **Fernández-Valmayor, A.** V Jornadas del Campus Virtual de la UCM: Buenas prácticas e indicios de calidad. [En línea] 2009. [Citado el: 11 de agosto de 2011.] <http://eprints.ucm.es/9689/>.
- [95]. **Chickering, A. W. y Gamson, Z. F.** Seven Principles For Good Practice in Undergraduate Education. [En línea] 1987. [Citado el: 11 de agosto de 2011.] <http://learningcommons.evergreen.edu/pdf/fall1987.pdf>.
- [96]. **Bunge, M.** La Ciencia. Su método y su filosofía. [En línea] 1958. [Citado el: 04 de septiembre de 2011.] http://www.aristidesvara.net/prb/metodologia/metodo_cientifico/naturaleza_metodo/bunge_libro_aristidesvara.pdf.
- [97]. **Benlloch-Dualde, J. V., Buendía, F. y Cano, J. C.** Tinta digital aplicada a la enseñanza universitaria de la Informática: menos lecciones y más actividades. [En línea] 2010. [Citado el: 06 de septiembre de 2011.] http://www.aenui.net/ActasJENUI/2010/Jenui2010_45.pdf.

- [98]. **Alur, D., Crupi, J. y Malks, D.** *Core J2EE Patterns, Best Practices and Design Strategies*. Segunda. s.l. : Prentice Hall, 2003.
- [99]. **Fernández-Valmayor, A. y otros.** Factores Claves para el éxito de un Gran Campus Virtual: el Campus Virtual de la UCM. [En línea] 2010. [Citado el: 31 de mayo de 2011.] http://www.sistedes.es/TJISBD/Vol-4/No-3/articles/iselear10_submission_14.pdf. ISSN 1988-3455.
- [100]. **Navarro, A., Cristóbal, J., Fernández-Chamizo, C., Fernández-Valmayor, A.** *Architecture of a Multi-Platform Virtual Campus*. John Wiley & Sons, Ltd, 2011, Software: Practice and Experience. En prensa.
- [101]. **Cristóbal, J., Merino, J., Navarro, A., Peralta, M., Roldán, Y., Silveira, R.** *Software Engineering Infrastructure in a Large Virtual Campus*. Interactive Technology and Smart Education. En prensa.