

Recepción: 15/07/2015  
Evaluación: 17/07/2015  
Aprobación: 08/09/2016  
Artículo de Reflexión

## UNA APROXIMACIÓN HACIA LOS PATRONES PEDAGÓGICOS ORIENTADOS HACIA DIDÁCTICAS EN COMPUTACIÓN

**Jesús Insuasti Portilla**

Universidad de Nariño

*Grupo de Investigación: Galeras.NET*

*insuasty@udenar.edu.co*

### RESUMEN

Dentro de la búsqueda constante de prácticas apropiadas para la enseñanza de los cursos de ciencias de la computación, un trabajo de investigación fue realizado a fin de construir una didáctica específica al curso de diseño de software donde fueron consultadas 19 universidades (15 internacionales de Estados Unidos y Europa; y 4 nacionales) en el mundo. Los profesores a cargo del curso de diseño de software de dichas universidades realizaron aportes en ocho dimensiones que fueron formuladas dentro de tal investigación.

Con estos valiosos aportes, se logró determinar un conjunto de elementos comunes –sin importar las diferencias culturales– de prácticas que fueron destacadas como exitosas desde la perspectiva de los docentes. El “producto” final de la mencionada investigación ha generado un marco teórico de trabajo sobre los lineamientos generales en la formulación de didácticas en computación. Sin embargo, la didáctica propuesta no ha sido probada en escenarios reales hasta ahora.

Dada esta situación, el presente artículo de reflexión pretende explorar los escenarios reales donde dichas prácticas de enseñanza sean factibles de llevar a cabo. Para tal menester, un concepto relativamente nuevo se introduce a fin de llevar a la práctica los resultados de la investigación en cuestión; este concepto se refiere a los patrones pedagógicos.

**Palabras clave:** didácticas, computación, patrones pedagógicos.

## AN APPROACH TO TEACHING COMPUTER-ORIENTED PEDAGOGICAL PATTERNS

### ABSTRACT

In the constant search for appropriate practices for teaching Computer Science courses, research was carried out in order to build a specific didactic Software Design course where 19 universities in the world were consulted (15 international—in the United States and Europe, and 4 national universities). Teachers in charge of the Software Design courses of these universities made contributions in eight areas that formulated the research.

The accomplishments of these valuable contributions were used to determine a set of common elements—regardless of cultural differences—in practices that were deemed successful from the perspective of the teachers. The final “product” of the research generated a theoretical framework on the general guidelines in the formulation of educational computing. However, the proposed method of teaching has not been tested in real-world scenarios so far. Given this situation, the article reflects upon the aims to explore real-world scenarios where such teaching practices are feasible.

For this task, a relatively new concept is introduced to implement the results of the investigation; this concept refers to the pedagogical patterns.

**Keywords:** teaching, computing, teaching patterns

## UMA APROXIMAÇÃO CAMINHO AOS PADRÕES PEDAGÓGICOS GUIADOS NAS DIDÁTICAS NA COMPUTAÇÃO

### RESUMO

Dentro da busca constante de cursos de ensino apropriados nas práticas de ciência da computação, foi realizada uma pesquisa a fim de construir uma didática específica para o desenho software onde foram consultados 19 universidades (internacionais 15 de estados unidos e Europa e 4 nacionais) no mundo. Os professores encarregados do curso de desenho de software dessas universidades fizeram contribuições em oito dimensões que foram formuladas em tais pesquisas.

Com estas contribuições valiosas, determinou-se um conjunto de elementos comuns, sem importar as diferenças culturais de práticas que foram destacadas como bem sucedida do ponto de vista dos professores. O “produto” final de que a investigação tem gerado um quadro teórico sobre de trabalhos sobre as orientações gerais na formulação de computador educacional. No entanto, a proposta didáctica não foi testado em cenários reais até agora.

Perante esta situação, este artigo explora os cenários de reflexão reais onde essas práticas de ensino são viáveis para levar a cabo. Para essa necessidade, um conceito relativamente novo é introduzido a fim de levar À pratica os resultados da investigação em questão; Este conceito refere-se a os padrões pedagógicos.

**Palavras-chave:** didático, computação, padrões pedagógicos

## INTRODUCCIÓN

Es evidente que la producción intelectual de las disciplinas asociadas a la computación es prolifera; en materia tecnológica, la producción derivada del avance del conocimiento en computación se realiza en periodos de tiempo bastante cortos –por lo general se mide en meses e incluso semanas–. Teniendo en cuenta este escenario de construcción dinámica y permanente del conocimiento, las disciplinas relacionadas a la computación se enfrentan a un vertiginoso proceso de actualización en contenidos desde el punto de vista de la formación académica. Al tenor de esta situación, quizás no sea la permanente actualización disciplinar la mayor preocupación desde una perspectiva educativa, sino la forma en que se debe abordar las dinámicas de enseñanza de dicho conocimiento disciplinar.

El presente artículo realiza una reflexión sobre los escenarios factibles donde podrían ser implementadas algunas experiencias que han sido consideradas por algunos profesores en computación alrededor del planeta; dichas experiencias en enseñanza han sido recopiladas a través de una investigación sobre la construcción de la didáctica específica para el curso de diseño de software, trabajo que dio origen a la publicación de un libro de lineamientos de didácticas en computación. Sin embargo, las experiencias que fueron consideradas exitosas por parte de los profesores que participaron en dicha investigación no han sido probadas en conjunto. Por tal razón, el artículo propone espacios donde dichas experiencias exitosas podrían implementarse a través de un concepto relativamente nuevo llamado patrones pedagógicos.

Basándose en los resultados de la investigación sobre la Didáctica del Diseño de Software<sup>1</sup> que fue producida dentro del Programa de Maestría en Docencia Universitaria de la Universidad de Nariño, la aplicación de las estrategias de enseñanza pueden ser reorientadas hacia la implementación de patrones pedagógicos como constructos teóricos que permiten capturar conocimiento experto en un determinado dominio. De esta forma, el presente artículo realiza una reflexión sobre la posible adaptación de los resultados de la investigación dentro del marco de trabajo de dichos patrones a fin de visualizar un campo de aplicación en escenarios reales.

Es el momento de vislumbrar espacios donde se ponga en práctica la teoría generada a través de la investigación en mención; para ello, la implementación

---

1. Investigación realizada en el Programa de Maestría en Docencia Universitaria, adscrita a la línea de investigación de Pedagogías y Didácticas en la Educación Superior; inicio: Septiembre de 2007; fin: Junio de 2010.

de patrones pedagógicos parece ser una de las alternativas más adecuadas en el campo de la enseñanza de las ciencias de la computación y disciplinas afines.

## 1. Elementos comunes de la enseñanza en computación

La investigación en cuestión presentó como resultado la creación de una didáctica específica para el curso de diseño de software en las disciplinas relacionadas al área de la computación. Realizando una síntesis sobre las conclusiones de la investigación, se puede observar un resumen a través de la tabla 1.

**Tabla 1. Matriz de características de experiencia en la enseñanza de diseño de software**

Dimensión	Características
Enfoque pedagógico	<ul style="list-style-type: none"> <li>- Las intenciones son conductistas y progresistas.</li> <li>- Se prefiere el modelo progresista por el método, metas, concepto de desarrollo, relación profesor-alumno, y evaluación de la formación.</li> <li>- Los contenidos evidencian el modelo conductista.</li> </ul>
Objetivos pedagógicos	<ul style="list-style-type: none"> <li>- Muchos profesores hacen explícitos los objetivos en los syllabus.</li> <li>- La redacción apunta hacia objetivos y metas, mas no hacia competencias.</li> </ul>
Contenidos	<ul style="list-style-type: none"> <li>- UML como herramienta principal.</li> <li>- Temáticas propuestas por ACM Computing Curricula.</li> </ul>
Actividades de clase	<ul style="list-style-type: none"> <li>- Preferencia marcada por la cátedra magistral.</li> <li>- Escasas actividades donde el estudiante es el actor principal.</li> </ul>
Actividades de trabajo independiente	<ul style="list-style-type: none"> <li>- Preferencia marcada por tareas y lecturas complementarias.</li> <li>- Escasa utilización de actividades y herramientas on-line.</li> </ul>
Recursos educativos	<ul style="list-style-type: none"> <li>- Uso de documentación impresa y digital.</li> <li>- Recursos tradicionales para el desarrollo de la cátedra en el aula.</li> </ul>
Espacio físico	<ul style="list-style-type: none"> <li>- Uso del salón de clase para desarrollo teórico.</li> <li>- Uso de los laboratorios para desarrollo práctico.</li> </ul>

Fuente: Jiménez, Martínez, & Insuasti, 2010: 96-97.

De lo anterior se concluye que los modelos pedagógicos predominantes en el desarrollo del curso de diseño de software de las 19 instituciones de educación superior que fueron consultadas son de corte conductista, basándose en la metodología de enseñanza instruccional. De esta forma, se plantean nuevos referentes a partir de dicha investigación con el fin de promover prácticas de aprendizaje activo. La propuesta final que es producto de la labor de la mencionada investigación articula principios del aprendizaje activo donde se otorga mayor participación en la construcción del saber dentro del aula por parte de los estudiantes.

La investigación presenta una serie de elementos comunes sobre las prácticas reconocidas como exitosas por parte de los profesores, las cuales giran alrededor de las ocho dimensiones expuestas en la tabla 1. De esta

manera, se establecen un conjunto de formatos estándar a fin de organizar –y dejar registro escrito– sobre las actividades desarrolladas tanto en aula como en los espacios del trabajo independiente. Así, se constituye en una bitácora del progreso académico donde se hace un seguimiento sobre los procesos de enseñanza y aprendizaje llevados a cabo dentro del curso de diseño de software; adicionalmente se adopta el desarrollo de proyectos de aula con fines prácticos con el propósito de promover espacios de construcción colectiva dada por el trabajo en equipo.

La didáctica construida describe detalladamente los diferentes aspectos de como puede ser implementada, desde la planeación de la clase y las actividades de trabajo independiente, hasta los procesos de evaluación. Terminada dicha investigación, se realiza la recomendación sobre la posible puesta a prueba de la didáctica en ambientes reales. Este es algo que aun falta por cumplir dado que la validación de la didáctica no formaba parte de los objetivos de la investigación que la produjo (Jiménez, Martínez, & Insuasti, 2012: 136). Entonces, la validación de las estrategias de enseñanza en contextos reales sin importar las diferencias culturales es de gran interés para el desarrollo de nuevos proyectos de investigación. El presente artículo realiza la reflexión sobre los aspectos a tenerse en cuenta para la implantación de didácticas en computación con un enfoque especial basado en patrones pedagógicos; temática que se aborda a continuación.

## **2. Una mirada hacia los patrones pedagógicos**

Históricamente, las disciplinas asociadas a la computación han sido clasificadas siguiendo los lineamientos de ACM, IEEE y AIS en su declaración llamada Computing Curricula (2005). Dentro de dichas disciplinas, una de ellas tiene por objeto de estudio la construcción de software bajo criterios de calidad; esta es la ingeniería de software. Con esta breve sinopsis sobre el área de la computación, el término “patrón” fue acuñado a partir del concepto de patrón de diseño que esta orientado hacia la concepción de una solución a un problema recurrente en un contexto concreto (Alexander et al, 1977).

Dado su origen en la ingeniería de software, un patrón de diseño es un modelo cuya abstracción se enmarca dentro de parámetros generales y ajustables a diferentes contextos con características comunes. De esta manera, un patrón de diseño es visto como una solución común para ciertos problemas relacionados con la producción de software (Sommerville, 2011: 190). Teniendo en cuenta los resultados de la investigación sobre la construcción de una didáctica específica aplicada hacia el diseño de software, elementos comunes fueron detectados en materia de enseñanza sin importar los contextos culturales. Dichos elementos comunes en materia de prácticas de enseñanza pueden ser concebidos como “soluciones comunes” aplicadas a entornos de enseñanza aprendizaje. Sin el ánimo de encasillar dicha orientación dentro de una postura de pensamiento

de radical positivismo, la concepción de “solución común” debe ser abordada desde su naturaleza compleja en materia de impacto en la pequeña sociedad del aula y no como una estructura inamovible de racionalidad instrumental.

Con este panorama, varios autores han vislumbrado la posible articulación del término “patrón” –muy propio de la ingeniera de software– con los ambientes de enseñanza dentro de los escenarios de la academia en computación. En síntesis, los patrones pedagógicos en la educación son concebidos como modelos para tratar el problema de cómo enseñar (Fricke & Völter, 2000). La idea del uso de patrones en pedagogía como un medio para presentar la experiencia docente fue propuesta inicialmente por Lilly (1996) en el sentido de implementar patrones reutilizables de diseño pedagógico orientado a la enseñanza de en general; ergo, los patrones pedagógicos de esta forma serían altamente adaptables para cualquier escenario académico no solamente para las disciplinas asociadas a la computación.

En consecuencia, los patrones pedagógicos deberán tener las siguientes características: deben ser repetibles y fácilmente adaptables a diferentes contextos, deben describirse de una manera que les permite ser fácilmente aplicados a pesar de las diferencias culturales, y dichos patrones deben comunicar soluciones ampliamente probadas a problemas comunes. En conclusión, un símil interesante es expuesto cuando se pretende articular técnicas de ingeniería de software a procesos educativos, en especial a procesos de enseñanza y aprendizaje; *“Los patrones pedagógicos juegan el mismo papel en la enseñanza y el aprendizaje que los patrones de diseño en el desarrollo de software”* (Seffah y Grogono, 2002: 18).

*“Los patrones pedagógicos tienen una serie de cualidades que, combinadas, brindan el potencial de ser una forma efectiva de compartir la experiencia de aprendizaje en red”* (Goodyear et al., 2004). También ha sido señalado por Goodyear que el enfoque de los patrones pedagógicos *“puede ser visto como una forma de puente entre la filosofía de enseñanza, los valores, la teoría, la evidencia empírica y experiencia (por un lado) y el contexto de la enseñanza práctica de diseño”* (2005: 92). La mayoría de los estudios de investigación en esta área se centran en destacar la eficacia de los patrones pedagógicos para los profesores noveles. En el contexto de la enseñanza de los cursos de las disciplinas asociadas a la computación, es importante hacer hincapié en que no sólo puede efectivamente ayudar a los profesores noveles en la enseñanza, sino que también puede ayudar al profesor con experiencia para comprender mejor la filosofía de un nuevo entorno de enseñanza como el que brinda el aprendizaje activo y colaborativo (Wen, Looi, & Chen, 2012: 141).

Para lograr la articulación entre el diseño estructurado y el enfoque pedagógico, los patrones pedagógicos llevan implícito el uso de un lenguaje particular en su construcción. Es así como un lenguaje de patrones es utilizado en el diseño y descripción de las experiencias en enseñanza que son consideradas como relevantes para su reutilización. Sin embargo, el uso

de una nomenclatura específica ha sido la causa para que dichos patrones pedagógicos sean diseñados y reutilizados específicamente en el área de la computación y disciplinas afines, puesto que no todos los educadores tienen un grado de formación en este particular.

Los patrones pedagógicos pretenden “*capturar el conocimiento experto de la práctica de la enseñanza y el aprendizaje*” a través de un diseño compacto y fácil de comunicar (Pedagogical Patterns Project, s.f.). Aunque los patrones pedagógicos han existido desde 1995 (Eckstein, 2001), algunos autores afirman que su impacto es limitado. Fincher y Utting afirman que “*los patrones pedagógicos aún carecen de una aceptación generalizada*” (2002: 201). Esta declaración es compatible con las observaciones hechas por Bennedsen y Eriksen, quienes dicen que “*A nuestro parecer, no existen informes sobre la conciencia de los enfoques pedagógicos de los profesores, ninguna evaluación sistemática de los efectos de los patrones pedagógicos u otras formas de la evaluación de los patrones pedagógicos*” (2003: T4A-3). Sin embargo, ambas afirmaciones se basan en las observaciones de los autores y no en evaluaciones metódicas en el uso y difusión de las pautas pedagógicas. Teniendo en cuenta lo anterior, es preciso indagar sobre las características propias de los patrones pedagógicos enmarcados dentro de una corriente epistemológica o corriente de pensamiento. Al parecer, todo tiende hacia un enfoque de aprendizaje activo el cual pertenece a la corriente constructivista (Active Learning, s.f.).

Actualmente existe mundialmente un proyecto común donde se realizan aportes en el ámbito de los patrones pedagógicos; dicho proyecto es liderado por el profesor Joseph Bergin –profesor de ciencias computacionales de Pace University en la ciudad de Nueva York, USA–, junto con aportes relevantes de diferentes universidades norteamericanas. Dentro de la zona de los patrones pedagógicos, varios lenguajes de patrones se han desarrollado tal como lo declara el proyecto internacional. Los lenguajes de patrones presentados en la página web de *Pedagogical Patterns Project* se organizan de diferentes maneras: algunos se organizan de acuerdo a una determinada actividad (por ejemplo, la retroalimentación), otros de acuerdo a los valores pedagógicos (por ejemplo, el aprendizaje activo). Esta diversidad de organización hace que sea difícil para los profesores el hecho de seleccionar un lenguaje de patrones y el patrón apropiado cuando se enfrentan a un problema de enseñanza concreto. En el campo del diseño de software orientado a objetos de la noción de patrones tenido un gran avance en 1995 gracias al éxito y la apropiación que ha tenido la producción escrita denominada “*the gang of four book*” (Gamma, Helm, Johnson, y Vlissides, 1995).

### **3. Implementación de patrones pedagógicos**

Con el fin de implementar los patrones pedagógicos teniendo en cuenta los resultados de la investigación sobre la didáctica del diseño de software, es menester tener en cuenta una serie de conceptos que marcan el escenario de

despliegue. Dichos conceptos son: patrón de diseño, lenguaje de patrones, y sistema de patrones (Derntl & Botturi, 2006: 138).

Se debe entender por patrón de diseño a aquel modelo que transmite el núcleo de la solución a algún problema relevante y recurrente de diseño (Alexander et al., 1977). Para que el patrón como tal sea explícito en un determinado contexto, se debe crear una representación formalizada usando un lenguaje natural o artificial, incluyendo una orientación sobre por qué y cómo aplicar dicho patrón. El interés inicial con el que fue construida la teoría sobre patrones está dirigido hacia aquellos modelos relacionados con artefactos, es decir, patrones de diseño en el área de la ingeniería de software.

Por su parte, el lenguaje de patrones incorpora un conjunto de modelos correspondientes a un dominio específico del diseño, incluidas las normas para combinarlos, es decir, como las palabras y reglas gramaticales (Alexander et al., 1977). Los patrones son un idioma el cual se debe complementar de forma sinérgica, y al mismo tiempo proporcionar instrucciones y orientaciones sobre las combinaciones posibles. Cabe anotar que en la literatura se usan a veces el termino “colección de modelos” en lugar de lenguaje de patrones cuando se hace referencia a un conjunto de patrones sin supuestos adicionales sobre la estructura o las relaciones existentes entre ellos.

Finalmente, un sistema de patrones ofrece más que un lenguaje de patrones: ofrece herramientas para el uso creativo, es decir, maneras de crear nuevos modelos de comprensión de las raíces mismas del sistema *per-se* junto con las características del lenguaje. En los lenguajes naturales, esto es típico de los hablantes nativos: su competencia con su lengua materna les permite crear nuevas palabras y oraciones con un alto grado de comprensión. Desde la óptica de patrones pedagógicos, un sistema de patrones se refiere a un sistema conceptual –el cual estaría definido dentro de una corriente de pensamiento pedagógica y una didáctica claramente establecida–, que consiste en un conjunto de modelos reutilizables dentro de un dominio específico –entendiéndose como dominio específico, un contexto educativo en especial–.

Dado lo anterior, la implementación de patrones pedagógicos no es una tarea mecanicista ni instrumental; por el contrario, su articulación con entornos reales demanda una clara concepción y adopción de una corriente epistemológica dentro de un ambiente pedagógico. Los antecedentes sobre la aplicación de los patrones pedagógicos han sido realizados en las disciplinas asociadas a la computación; sin que esto quiera decir que la posibilidad de aplicación de los patrones pedagógicos sea exclusiva de esta área del conocimiento.

Durante el 2012, a través de los resultados de la investigación en mención, se ha venido desarrollando actividades conducentes a la reorganización de las prácticas de enseñanza catalogadas como exitosas por parte de los docentes



de las 19 universidades consultadas. En este sentido, es de vital interés la implementación de dichas experiencias en el marco de los patrones pedagógicos como técnica probada y comúnmente aceptada –por lo menos en el medio de la enseñanza de la computación y disciplinas afines–; así pues, se tendría que re-pensar sobre los lineamientos pedagógicos que dictan las políticas, acciones y comportamientos de los procesos educativos que están presentes en la academia. He aquí el primer problema que debería ser solucionado con alta prioridad, ¿Qué modelo pedagógico se está desarrollando en las diferentes universidades que imparten cursos de las ciencias computacionales y disciplinas a fines? La respuesta no es muy alentadora puesto que los resultados de la investigación sumados a otras fuentes de información indican que en la mayoría de los casos, el modelo pedagógico aparece como un ente sin cuerpo –aunque de antemano se conoce acerca de su naturaleza instruccionalista implícita–, y en otros casos, se marca la tendencia conductista.

## CONCLUSIONES

Siendo el presente artículo el comienzo para el desarrollo de una investigación de donde se busque validar la aplicación de estrategias de enseñanza en los cursos de computación y sus disciplinas asociadas, existe aún mucho camino por recorrer sobre las condiciones que se requieren para que la implementación de patrones pedagógicos pueda afianzarse en forma provechosa y productiva.

La tendencia general en materia de enseñanza en computación y disciplinas a fines es claramente marcada por los lineamientos de pedagogías conductistas y modelos instruccionales. Con la implementación de patrones pedagógicos dentro de una base sólida conceptual fundamentada por el aprendizaje activo, es posible explorar el potencial de corrientes epistemológicas que promueven la construcción participativa y desde un pensamiento crítico la “vida del aula”.

Los espacios de reflexión y construcción permanente sobre los proyectos de investigación del programa de Maestría en Docencia Universitaria de la Universidad de Nariño ha permitido formular, diseñar y desarrollar estudios de caso en la búsqueda de alternativas en materia de pedagogía y didáctica aplicadas a diferentes escenarios en la educación superior. Fiel reflejo del resultado del aprovechamiento de los espacios para la investigación y la crítica pueden evidenciarse en la producción del libro sobre construcción de didácticas específicas para computación, el cual está basado en las la investigación realizada al interior de la Maestría.

Elementos importantes de este trabajo son puntos de partida para la tesis doctoral asociada con el fenómeno de transposición didáctica y su potencial uso para diseños curriculares y generación de espacios de motivación del aprendizaje.

## REFERENCIAS BIBLIOGRÁFICAS

- Active Learning “Patterns for active learning” (s.f.). En Internet: <http://www.pedagogicalpatterns.org/current/activelearning.pdf>, Fecha de Consulta: Agosto 24, 2012.
- ACM, IEEE & AIS. Computing Curricula: The Overview Report 2005. En Internet: [http://www.acm.org/education/education/curric\\_vols/CC2005-March06Final.pdf](http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf), Fecha de Consulta: Septiembre 11, 2012.
- Alexander, C. et al. *A pattern language: towns, buildings, construction*. New York: Oxford University Press, 1977.
- Bennedsen, J., & Eriksen, O. “Categorizing Pedagogical Patterns by Teaching Activities and Pedagogical Values”, *Computer Science Education* (2009), 16(2), doi: 10.1080/08993400600768091, pp. 157-172.
- Bennedsen, J., & Eriksen, O. (2003) “Applying and developing patterns in teaching”, En: Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference. Champaign, IL: Stipes Publishing LLC.
- Bennedsen, J. (2006) “The Dissemination of Pedagogical Patterns”, *Computer Science Education*, 16(2), DOI: 10.1080/08993400600733590, pp. 119-136.
- Derntl, M., & Botturi, L. “Essential Use Cases for Pedagogical Patterns”, *Computer Science Education* (2006), 16(2), doi:10.1080/08993400600768182, pp. 137-156.
- Eckstein, J. “Pedagogical patterns: capturing best practices in teaching object technology”, *Software Focus*, 2(1) (2001).
- Fincher, S., & Utting, I. “Pedagogical patterns: their place in the genre”, En: *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education*. New York: ACM Press, 2002.
- Fricke, A., & Völter, M. *A pedagogical pattern language about teaching seminars effectively* (2000). En Internet: <http://www.voelter.de/data/pub/tp/html/index.html>, Fecha de consulta: Septiembre 2, 2012.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. *Design patterns: elements of reusable object-oriented software*. Reading, MA: Addison-Wesley, 1995.
- Goodyear, P. et al. “Towards a pattern language for networked learning”, En: *Goodyear, P. et al (Eds.), Networked learning*. Lancaster: Lancaster University, 2004.
- Goodyear, P. “Educational design and networked learning: Patterns, pattern languages and design practice”, *Australasian Journal of Educational Technology*, 21(1), (2005).
- Jiménez, R., Martínez, Á. & Insuasti, J. *Didáctica del Diseño de Software* (2010), En Internet: <http://biblioteca.udenar.edu.co:8085/bibliotecavirtual/viewer.aspx?&var=83284>, Biblioteca Alberto Quijano Guerrero: Universidad de Nariño, Pasto - Colombia.
- Jiménez, R., Martínez, Á. & Insuasti, J. *Lineamientos para una didáctica en computación - Caso: Curso de Diseño de Software*. Alemania: Editorial Académica Española, en colaboración con AV Akademikerverlag GmbH & Co., 2012.
- Lilly, S. “Patterns for pedagogy”, *Object Magazine*, 5(8), (1996).
- Pedagogical Patterns Project* (s.f.) En Internet: <http://www.pedagogicalpatterns.org>, Fecha de Consulta: Agosto 25, 2012.
- Seffah, A., & Grogono, P. *Learner-centred software engineering education: From resources to skills and pedagogical patterns* (2002). En: T. Leth-bridge, W. M. McCracken & M. Lutz (Eds.), Proceedings of the 15th Conference on Software Engineering Education and Training. Los Alamitos (CA): IEEE Computer Society Press.
- Sommerville, I. *Software Engineering 9th Edition*. Boston (MA), USA: Pearson/Addison-Wesley, 2011.
- Wen, Y., Looi, C.-K., & Chen, W. “Supporting Teachers in Designing CSCL Activities: A Case Study of Principle-based Pedagogical Patterns in Networked Second Language Classrooms”, E (2012).