

# Metasíntesis sobre la aplicación de principios de Ingeniería de Software en el desarrollo de plataformas de tecnología educativa

## Meta-synthesis regarding the application of Software Engineering principles on the development of educational technology platforms

Mauricio Arturo Ibarra-Corona 

Universidad Autónoma de Querétaro (México)  
[mauricio.ibarra@uaq.mx](mailto:mauricio.ibarra@uaq.mx)

Alexandro Escudero-Nahón 

Universidad Autónoma de Querétaro (México)  
[alexandro.escudero@uaq.mx](mailto:alexandro.escudero@uaq.mx)

Recibido: 13/01/2021

Aceptado: 14/05/2021

Publicado: 01/06/2021

### RESUMEN

Debido a la creciente presencia de la tecnología digital en los entornos educativos formales, las aplicaciones digitales que apoyan los procesos de enseñanza-aprendizaje son cada día más sofisticados y los docentes están tomando parte activa en el diseño de esas aplicaciones. Uno de los aspectos que más ha evolucionado es el desarrollo de software para el diseño de plataformas de tecnología educativa. Este aspecto ha sido motivo de diversas investigaciones científicas, pero no existen publicaciones que den cuenta de cómo han sido considerados los principios de la Ingeniería de Software por parte de los docentes en el desarrollo de software para el diseño de plataformas de tecnología educativa. Para cumplir con lo anterior, se realizó una revisión sistemática de la literatura especializada publicada en los últimos cinco años con el método de investigación documental propio de la metasíntesis. La obtención de información se realizó en las bases de datos científicos *Springer Link*, *Science Direct*, *ERIC* y *CONRICyT* con la siguiente fórmula: "Software Engineering" AND ("Instructional Design" OR "Educational Technology"). Fueron analizados en total 69 artículos escritos en inglés o español. Tras una interpretación hermenéutica de los resultados, el hallazgo más relevante sugiere que, aunque los principios de Ingeniería de Software sí son contemplados y aplicados por la mayoría de los docentes, existe una brecha entre la teoría y la práctica referente a la tecnología educativa, misma que deriva de la complejidad de empatar la pedagogía con el desarrollo tecnológico. Finalmente, se sugiere el desarrollo de un modelo que facilite aplicar los principios de la Ingeniería de Software en el proceso de diseño de las plataformas, que a su vez facilitaría el proceso de desarrollo de software educativo.

### PALABRAS CLAVE

Tecnología Educativa; Integración Tecnológica; Tecnologías de la Información; Sistemas de Información; Diseño Instruccional.

### ABSTRACT

Due to the growing presence of digital technology in formal educational environments, digital applications that support teaching-learning processes are becoming more sophisticated and teachers are taking an active part in its design. One of the aspects that has evolved the most is the development of software for the design of educational technology platforms. This aspect has been the subject of various scientific research, but there are no publications that account for how the principles of software engineering have been considered by teachers in the development of software for the design of educational technology

platforms. To comply with the above, a systematic review of the specialized literature published in the last five years was carried out with the documentary research method of meta-synthesis. The information was obtained in the scientific databases *Springer Link*, *Science Direct*, *ERIC* and *CONRICyT* with the following formula: "Software Engineering" AND ("Instructional Design" OR "Educational Technology"). A total of 69 articles written in English or Spanish were analyzed. After a hermeneutical interpretation of the results, the most relevant finding suggests that, although the principles of software engineering are contemplated and applied by most of the teachers, there is a gap between theory and practice regarding educational technology, which derives from the complexity of matching pedagogy with development. Finally, the development of a model that facilitates the application of software engineering principles in the platform design process is suggested, which in turn would facilitate the educational software development process.

## KEYWORDS

Educational Technology; Technology Integration; Information Technology; Information Systems; Instructional Design.

## CITA RECOMENDADA

Ibarra-Corona, M.A. & Escudero-Nahón, A. (2021). Metasíntesis sobre la aplicación de principios de Ingeniería de Software en el desarrollo de plataformas de tecnología educativa. *Revista Interuniversitaria de Investigación en Tecnología Educativa*, 10, 62-75. <https://doi.org/10.6018/riite.463421>

## Principales aportaciones del artículo y futuras líneas de investigación:

- Identificación sobre cómo han sido aplicados los principios de Ingeniería de Software en el diseño de plataformas de tecnología educativa por parte de los docentes.
- Método de investigación documental mediante referencias de una revisión sistemática.

## 1. INTRODUCCIÓN

A medida que el uso de la tecnología digital aumenta en los ámbitos educativos, las áreas de oportunidad, así como el potencial para la enseñanza y el aprendizaje no dejan de crecer. Sin embargo, los rápidos avances tecnológicos presentan nuevos problemas para el desarrollo de software educativo (García et al., 2009). Lo anterior es una labor compleja porque determinar si el proceso que se sigue, si los componentes y si los objetos desarrollados cumplen con las expectativas o requerimientos de los estudiantes y profesores siempre implica una consideración pedagógica. De tal manera, la constante evolución de la educación demanda el diseño y personalización de tecnologías educativas con requerimientos diversos y de gran escala (Chimalakonda & Nori, 2020).

Desde la perspectiva de la Ingeniería de Software, cuando se trata de un desarrollo tecnológico, es necesario seguir o diseñar algún proceso que permita asegurar la calidad del producto, para así cumplir con las expectativas derivadas del mismo (Sengupta & Dasgupta, 2017b). Sin embargo, aunque lo anterior también es cierto para el diseño y desarrollo de la tecnología educativa, es difícil hablar de una única metodología que permita satisfacer necesidades tan diversas (Doderó et al., 2012), puesto que los procesos involucrados en la enseñanza y el aprendizaje siempre están definidos por diversas variables.

La Ingeniería de Software, sin embargo, no ofrece una única metodología, sino múltiples modelos de referencia y metodologías con el objetivo de proveer a los desarrolladores de opciones que puedan aplicar según su criterio y el contexto del proyecto (Jabangwe et al., 2018). Las metodologías pueden ser clasificadas en dos categorías: tradicionales, que se refieren a la

existencia de un proceso formal, estricto e inflexible; y ágiles, donde, si bien existe un proceso definido, no es indispensable seguirlo de manera puntual. Sin importar de que categoría se trate, las metodologías, no son más que un conjunto de buenas prácticas y de principios, que pueden ser utilizados según sea conveniente para el proyecto en turno (Dybå & Dingsøy, 2008).

Entonces, resulta complejo determinar qué modelos y metodologías pueden ser utilizadas para cumplir con los objetivos deseados. Esto resulta aún más desafiante cuando se considera que, debido al creciente uso de entornos virtuales de aprendizaje y de sistemas de gestión de aprendizaje por parte de las instituciones educativas, se espera que la responsabilidad de implementar esas herramientas recaiga cada vez más sobre los docentes que las han de utilizar (Pástor et al., 2018).

Aunque el involucramiento de los docentes en el proceso de creación de tecnologías educativas es considerado como algo beneficioso (Kali et al., 2015), no deja de ser un hecho que no tienen una formación en el proceso de desarrollo y, por lo mismo, es posible que los principios y mejores prácticas de la Ingeniería de Software no se apliquen. No obstante, esta falta de formación funciona también en el sentido opuesto, en el entendido de que los ingenieros de software no cuentan con una formación pedagógica que los oriente al cumplimiento de las expectativas y necesidades presentes en los procesos de enseñanza-aprendizaje (Doderó et al., 2012).

Entonces, por una parte, existen principios, metodologías, modelos y buenas prácticas que la Ingeniería de Software ofrece para el desarrollo de tecnología educativa, mismos que han sido empleados de manera exitosa en software comercial y, por otra parte, existen todas las teorías sobre enseñanza, aprendizaje, diseño instruccional que siguen siendo motivo de estudio hasta la fecha, por lo que unir ambas disciplinas significa cerrar una brecha existente entre la teoría y la práctica de la tecnología educativa (Area-Moreira et al., 2016).

## 2. MÉTODO

La presente investigación documental tuvo por objetivo analizar cómo se han considerado y aplicado los principios de la Ingeniería de Software en el desarrollo de plataformas de tecnología educativa por parte de los docentes. Para cumplir con lo anterior, se realizó una revisión sistemática con el método de la metasíntesis sobre artículos cuya principal temática era el diseño y desarrollo de las plataformas. Este método de investigación fue el más idóneo debido al carácter cualitativo del tema de investigación. Además, porque la metasíntesis organiza los resultados de varios estudios en grupos homogéneos y es útil para sintetizar evidencia proveniente de diferentes tipos de documentación (cualitativa, cuantitativa o mixta) que puede ser obtenida en fuentes alternativas a las bases de datos científicos (Leary & Walker, 2018).

La obtención de la literatura especializada se realizó empleando los métodos propuestos por *The Campbell Collaboration*, a la par de algunas estrategias inusuales con la finalidad de incrementar el número de documentos sintetizados (Kugley et al., 2017; Thomas & Harden, 2008). La metasíntesis se desarrolló en cuatro fases (Sadelowski & Barroso, mencionado en Salinas & Marín, 2019).

### 2.1. Identificación del problema de investigación

Partiendo del objetivo planteado con anterioridad, la pregunta rectora que actuó como guía de la metasíntesis fue la siguiente: ¿cómo se han considerado y aplicado los principios de la Ingeniería de Software en el desarrollo de plataformas de tecnología educativa por parte de los docentes?

### 2.2. Recolección de estudios

La búsqueda de documentos se realizó en cuatro bases de datos científicas: *Springer Link*, *Science Direct*, *ERIC* y *CONRICyT*. Únicamente fueron contemplados artículos de investigación, ya que estos cuentan con un problema de investigación, un método de investigación, datos empíricos y un proceso de análisis. La búsqueda y recopilación de los documentos se realizó el

3 de septiembre de 2020 y el análisis e interpretación de los resultados se realizó desde esa fecha hasta noviembre de 2020.

Fueron considerados artículos de investigación publicados hasta cinco años previos a la fecha de búsqueda. Se definió un rango de búsqueda que fue desde septiembre del 2015 hasta septiembre de 2020. Fueron aceptados artículos escritos en inglés o español. Para las bases de datos *Springer Link*, *Science Direct* y *ERIC* se empleó la siguiente fórmula de búsqueda: "*Software Engineering*" AND ("*Instructional Design*" OR "*Educational Technology*"). Por su parte, en CONRICyT, se utilizó la siguiente fórmula: (Diseño de Software) AND (Tecnología educativa OR Diseño Instruccional), ya que los resultados más pertinentes se consiguieron al realizar la búsqueda en español (Gisbert & Bonfill, 2004; Kugley et al., 2017).

Se obtuvieron las siguientes cantidades de documentos por base de datos: *Science Direct* (101), *Springer Link* (263), *ERIC* (13), CONRICyT (19). En total, 396 artículos de investigación, mismos que fueron reducidos a 39 aplicando los criterios de inclusión y exclusión descritos a continuación.

Se consideraron como criterios de inclusión: que el documento fuese publicado entre septiembre de 2015 y septiembre de 2020; que fuera un artículo de investigación; que tratara sobre el diseño, desarrollo o implementación de una plataforma de tecnología educativa; y que hiciera referencia a la Ingeniería de Software como parte del proceso de desarrollo. Mientras que, como criterios de exclusión, se consideró: que se tratara de un estudio bibliométrico; que se tratara de un estudio teórico; o que implicara el desarrollo de metodologías, métodos o modelos referentes al proceso de enseñanza aprendizaje.

Considerando los pocos resultados pertinentes arrojados por las bases de datos, se determinó que sería necesario realizar una búsqueda alternativa en las referencias bibliográficas de cada uno de los 39 artículos admitidos para su posterior análisis. Tras realizar dicha búsqueda, aplicando los criterios de exclusión e inclusión, se obtuvieron otros 30 documentos, dando un total de 69 documentos que fueron empleados en fases posteriores.

Si bien los últimos documentos fueron obtenidos mediante un muestreo a conveniencia, es posible justificar esto debido a que la calidad de una metasíntesis se fundamenta en la triangulación de expertos del tema, fuentes de información y del contexto de las investigaciones admitidas. Mientras estas puedan ser rastreadas, la calidad no se ve afectada (Leary & Walker, 2018). El proceso de la búsqueda y recolección de artículos se presenta de manera sintetizada en la Figura 1.

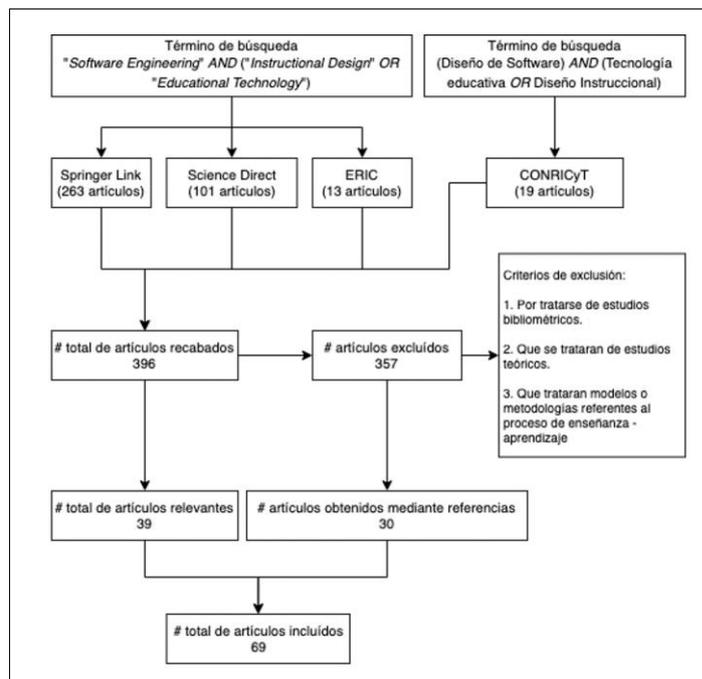


Figura 1. Proceso de recolección y evaluación de los artículos.

### 2.3. Evaluación de la calidad de los estudios seleccionados

Tras revisar a profundidad los 69 estudios admitidos para análisis, se determinó que todos cumplían con los criterios expuestos con anterioridad y responden la pregunta guía planteada en la primera fase de la metodología.

### 2.4. Codificación e interpretación de la información

Una vez que los 69 estudios se sometieron a todos los filtros pertinentes y fueron comparados tanto con la pregunta rectora como con los criterios de exclusión e inclusión, fue necesario establecer categorías de análisis que guiarán la síntesis de los documentos recolectados (Butler et al., 2018) y que sirvieron como etiquetas para clasificar las investigaciones y asignarlas para responder cada pregunta de investigación planteada (Salinas & Marín, 2019) (Tabla 1).

*Tabla 1. Categorías y preguntas temáticas*

Categoría	Pregunta
1. Aplicación de los principios de Ingeniería de Software.	¿Qué principios de Ingeniería de Software han sido aplicados en el diseño y desarrollo de plataformas de tecnología educativa por los docentes?
2. Proceso de diseño y desarrollo de las plataformas de tecnología educativa.	¿Qué proceso siguen los autores que han propuesto, diseñado y desarrollado plataformas de tecnología educativa?

Fuente: Elaboración propia

Definido lo anterior, los documentos fueron clasificados en la categoría más adecuada y pertinente. Se permitió que un mismo artículo estuviese presente en las dos categorías simultáneamente. Tras la clasificación, los resultados fueron interpretados y sintetizados, tal síntesis y clasificación se muestran en el siguiente apartado.

## 3. RESULTADOS

Los 69 estudios fueron clasificados en la categoría temática correspondiente según su capacidad de responder la pregunta de investigación asignada a cada categoría. Debido a las características de las investigaciones, se permitió que un mismo artículo perteneciera a ambas categorías. Se identificaron 42 artículos pertenecientes a la categoría 1 (Aplicación de los principios de Ingeniería de Software) mientras que, la categoría 2 (Proceso de diseño y desarrollo de las plataformas de tecnología educativa), se conformó de 40 estudios. En la Tabla 2, se presenta a detalle la clasificación categórica realizada.

*Tabla 2. Clasificación categórica de los estudios.*

Categoría	Cantidad
1. Aplicación de los principios de Ingeniería de Software.	29
2. Proceso de diseño y desarrollo de las plataformas de tecnología educativa.	27
3. Ambas.	13

Fuente: Elaboración propia.

A partir de la clasificación, se realizó la síntesis correspondiente a la vez que se buscó dar respuesta a la pregunta temática propia de cada categoría.

### 3.1. Aplicación de los principios de Ingeniería de Software

El rápido crecimiento del uso de las plataformas de tecnología educativa ha diversificado de manera significativa la manera en la que algunos autores aplican y consideran a la Ingeniería de Software (Dodero et al., 2012). Con la finalidad de organizar la información de cada uno de los

diferentes artículos sintetizados, se realizó la siguiente clasificación, en función de la aplicación de los principios de Ingeniería de Software en las plataformas de tecnología educativa: 1) Como elemento de desarrollo; 2) Como puente entre la tecnología y la pedagogía; 3) Como un proceso a partir del cual se obtienen las mejores prácticas; 4) Como principios de desarrollo; 5) Como patrones de diseño y arquitectónicos para plataformas de tecnología educativa; y 6) Como metodología de desarrollo de software.

La clasificación permite encontrar las similitudes y diferencias entre la aplicación que cada autor hace según su contexto. Esto, sin embargo, muestra la extensión y diversidad de la Ingeniería de Software, puesto que, según el propósito de la investigación, serían los principios que más aportarían a su cumplimiento.

Lo anterior se convierte en un común denominador de todas las investigaciones que no contemplaron a la Ingeniería de Software como una metodología de desarrollo de software. Esas investigaciones utilizan únicamente algunos principios, mas no buscan aplicar una metodología de desarrollo de software de manera estricta, puesto que proponen que hacer esto podría generar nuevos problemas, como relegar el aspecto pedagógico de la tecnología educativa, aumentando la brecha de conocimiento existente entre estos conceptos (Area-Moreira et al., 2016; Bennett et al., 2012; Squires & Preece, 1999; Yeoman & Ashmore, 2018).

Ahora bien, ¿cuál es la diferencia entre las seis categorías que tienen este común denominador?, la diferencia radica en qué principios consideran y cómo los emplean en su proceso de desarrollo, puesto que se puede considerar, desde un aspecto muy general, como un elemento en el proceso de desarrollo (Coomans & Lacerda, 2015; Golitsyna, 2015), o como una fuente de conceptos técnicos, necesarios para describir la estructura de la plataforma de tecnología educativa en su totalidad mediante patrones de diseño y patrones arquitectónicos, siendo ambos conceptos modulares de la Ingeniería de Software (Aguña, 2012; Campo et al., 2020; Chimalakonda & Nori, 2018; Devedžić & Jovanović, 2015; Kohen-Vacs et al., 2016; Marcolino & Barbosa, 2017; Pástor et al., 2018; Sengupta et al., 2009; Sengupta & Dasgupta, 2017a, 2017b).

Entonces, en un afán de buscar una sinergia entre lo que la Ingeniería de Software puede ofrecer (en forma de principios) y lo que la tecnología educativa requiere (en su proceso de desarrollo), se propuso tomar los principios que se necesitarían según el contexto y el propósito de la plataforma de tecnología educativa. Esto implicó integrar las mejores prácticas de la Ingeniería de Software en áreas como el *e-learning* y el diseño instruccional (Adnan & Ritzhaupt, 2018). Un claro ejemplo de esta propuesta, fue la de emplear los lenguajes de modelado de la Ingeniería de Software en el diseño de espacios virtuales de aprendizaje, lo que permite: 1) representar de manera gráfica la composición de estos espacios; 2) identificar los recursos pedagógicos; 3) analizar sus deficiencias y carencias; y 4) evaluar su calidad, usabilidad y utilidad (Chen et al., 2017; Damşa et al., 2019; Torres et al., 2014).

Así pues, considerar a la Ingeniería de Software como una fuente de principios y de mejores prácticas pareciera ser la solución a los problemas encontrados (como la brecha entre la teoría y la práctica en la tecnología educativa). De tal forma que se emplean aquellos que permiten lograr las características deseadas en las plataformas de tecnología educativa como lo son: usabilidad, utilidad, seguridad, confiabilidad, velocidad, disponibilidad y calidad (Aeiad & Meziane, 2019; Bettio et al., 2013; Chimalakonda & Nori, 2013; Frimpon, 2012; Mardis et al., 2018; Masson & Udas, 2009; Nori et al., 2014; Oyelere et al., 2018; Plaza et al., 2009; Zadahmad & Yousefzadehfard, 2016). No obstante, es necesario mencionar que todo lo descrito con anterioridad se trata de un escenario ideal, donde la aplicación de principios se podría dar de manera transparente y sin problemas, ya que no deja de ser una tarea compleja tratar de empatar ambas disciplinas debido a las diferencias conceptuales y semánticas existentes.

Es posible apreciar la dificultad de dicha integración en los trabajos de investigación que reconocen el papel de la Ingeniería de Software, pero no es aplicada o se propone como trabajo a futuro. De esta manera, queda patente que, aunque es del conocimiento de los docentes, prefieren no utilizarla debido a que no encuentran cómo emplearla de manera efectiva y sin caer de nueva cuenta en la brecha descrita con anterioridad (Chimalakonda & Nori, 2012; Coutinho & Bezerra, 2020; Dehcordi & Alavi, 2019; Kale et al., 2020).

Así pues, en respuesta a la pregunta temática, ¿qué principios de Ingeniería de Software han sido aplicados en el diseño y desarrollo de plataformas de tecnología educativa por los docentes? Se encontró que se trataban de principios orientados a la mejora de procesos de desarrollo de software tales como: eficiencia, agilidad, reusabilidad, patrones de diseño, patrones arquitectónicos y modelación (Chimalakonda & Nori, 2020; Garcia et al., 2009; González-González et al., 2015; Zhang & Dorn, 2012), mismos que buscarían ser empatados y adaptados para su aplicación conjunta con la teoría sobre tecnología educativa, labor que hasta la actualidad ha resultado compleja debido a la brecha existente entre la teoría y la práctica en la tecnología educativa.

### **3.2. Proceso de diseño y desarrollo de las plataformas de tecnología educativa**

El desarrollo de plataformas de tecnología educativa involucra invariablemente un proceso, con requerimientos específicos derivados de su carácter pedagógico e instruccional. Por su parte, la Ingeniería de Software es un área del conocimiento tan extensa, que su definición, principios y aplicación rara vez son homogéneos (Alharthi et al., 2019; Muyinda, 2007). Es así como, al analizar la literatura especializada, se encontraron distintos términos y aplicaciones, mismos que dependían del contexto en el que cada autor desarrolló su investigación, haciendo complicado responder a la pregunta: ¿Qué proceso siguen los autores que proponen, diseñan y desarrollan plataformas de tecnología educativa?

Los procesos de diseño y desarrollo seguidos en las investigaciones fueron heterogéneos. Algunas siguieron un proceso estricto, aplicando una metodología de desarrollo de software en forma, mientras que otras obedecieron un proceso propio, sin contemplar en ningún momento los principios y buenas prácticas que la Ingeniería de Software ofrece. En este entendido, se determinó que el proceso a seguir se veía definido por tres factores: 1) conocimientos técnicos del autor; 2) extensión y alcance de la plataforma de tecnología educativa desarrollada; y 3) área del conocimiento del autor (Torres et al., 2014; Yulianto et al., 2015). Tiene sentido que la formación del docente tenga un impacto directo sobre la realización de su trabajo. Es coherente que un profesional con conocimientos sobre tecnologías de la información, sistemas computacionales o Ingeniería de Software tendrá una predisposición por utilizar modelos, herramientas y metodologías de desarrollo de software.

Lo anterior, tiene aún más sentido cuando se analiza el proceso que cada autor siguió o propuso para diseñar y desarrollar plataformas de tecnología educativa. Se encontró que los procesos que se siguen se podrían clasificar en tres categorías: 1) procesos formales, con fundamento en las metodologías de desarrollo de software; 2) procesos mixtos, con fundamento en los principios de Ingeniería de Software y de tecnología educativa; y 3) procesos propios, que no tienen fundamento en las metodologías de desarrollo de software.

En la primera categoría, el proceso seguido obedeció a metodologías de desarrollo de software formales, con autores que incluso hacen la distinción entre las tradicionales (como el proceso racional unificado) (Yulianto et al., 2015) y las ágiles, llamadas así por su eficiencia, flexibilidad y velocidad de desarrollo (Arimoto et al., 2016). Aquí, los procesos tanto de diseño como de desarrollo se realizan de forma iterativa, construyendo componentes que puedan ser reutilizables y que sean fáciles de mantener y modificar. Al mismo tiempo, se contemplan evaluaciones y pruebas para el aseguramiento de la calidad (Bakki et al., 2020; McEneaney, 2016) (propias de la Ingeniería de Software), con el añadido del aseguramiento de la calidad del aprendizaje, indispensable en la tecnología educativa.

Como contraste, en los procesos mixtos, con fundamento en los principios de Ingeniería de Software y de tecnología educativa, aunque no se emplea ninguna metodología formal, sí se reconoce la necesidad de una guía para el proceso de diseño y desarrollo (Crescenzi-Lanna & Grané-Oró, 2016; Martin et al., 2011; Setirek & Tanrikulu, 2015). Lo anterior, es el resultado de la inclusión de los docentes en el proceso de diseño de las plataformas (Ertmer, 1999; Kali et al., 2015). Es importante mencionar que, en esta categoría, sí se aplican principios de Ingeniería de Software, describiendo incluso las etapas generales por las que todo proceso de desarrollo de software debería de pasar (Oyelere & Suhonen, 2016). Incluso se presentaron los criterios que determinan el éxito o fracaso de una plataforma de tecnología educativa (Sun et al., 2008). Como último punto importante de esta categoría, se planteó la idea de que los ingenieros de software

deberían incluirse en el proceso de diseño instruccional, incluso con docentes que, aunque reconocen la falta de una aplicación de un proceso formal, contemplan el proceso de Ingeniería de Software (Alzahrani et al., 2018; Crowe et al., 2017; Douglas, 2001; González-Martínez et al., 2015; Meza Cano et al., 2016; Roschelle et al., 1999; Tabares Morales et al., 2017).

Por último, en los procesos propios, ningún docente contempló algún principio sobre Ingeniería de Software, modelo o metodología de desarrollo. En su lugar, emplearon un proceso de diseño y desarrollo guiado por la retroalimentación de los usuarios finales del sistema (estudiantes y docentes) (Bedregal-Alpaca et al., 2019; Gašević et al., 2017) o por la efectividad que la plataforma tuvo en la mejora del proceso de enseñanza-aprendizaje (Chounta et al., 2017; Flores-Fuentes & Juárez-Ruiz, 2017). Por otra parte, en algunos casos se optó por identificar los componentes de la plataforma que deberían mejorarse, desarrollarse o incluirse. Sin embargo, todo desde un punto de vista pedagógico, sin tomar en consideración las implicaciones que las mejoras o modificaciones tendrían en el sistema (Osuna-Acedo et al., 2018; Richardson et al., 2020; Ritzhaupt & Kumar, 2015; Rook et al., 2015; Vega López et al., 2019).

#### 4. DISCUSIÓN Y CONCLUSIONES

Cuando se aplica la Ingeniería de Software al diseño y desarrollo de plataformas educativas se adaptan los principios propios de esta ingeniería según el criterio de cada docente. En este sentido, cada docente utiliza aquello que considera más conveniente y que le permite cumplir los objetivos de su proyecto. Resulta interesante que términos y conceptos propios de los procesos de desarrollo de software les son familiares a los docentes. Es remarcable cómo logran hacer ciertas analogías entre los procesos de diseño instruccional y de desarrollo de software (Chimalakonda & Nori, 2018; Douglas, 2001). Es así como es posible evidenciar la capacidad que tiene la Ingeniería de Software de adaptarse y diversificarse en distintas áreas del conocimiento (Adnan & Ritzhaupt, 2018). Sin embargo, es evidente que existe una brecha en el conocimiento entre los fundamentos teóricos referentes a la tecnología educativa y la puesta en práctica de estos.

El hecho de que aquellos que diseñan y proponen las bases teóricas sean los mismos que implementen y desarrollen herramientas y plataformas de tecnología educativa tiene beneficios. Por ejemplo, el que se encuentre presente un sustento pedagógico detrás de cada componente orientado al aprendizaje (Campo et al., 2020; Marcolino & Barbosa, 2017). No obstante, cuando el proceso que se sigue para desarrollar la plataforma no es el adecuado o se aplica de forma errónea, se presentan errores en el producto final. Lo anterior tiene implicaciones significativas en la percepción de uso de la plataforma, así como en su capacidad de captar y mantener a los estudiantes. Es indiscutible que estas últimas cualidades sirven como métricas de la calidad del software y cuando esta es baja, el software rara vez tiene éxito (Aeiad & Meziane, 2019; Plaza et al., 2009).

Es notable que en aquellas investigaciones en las que se desarrolla software, nunca se miden aspectos cualitativos del mismo, como lo son: usabilidad, experiencia de usuario, escalabilidad, extensibilidad, seguridad y portabilidad. Únicamente se evalúan aspectos referentes al aprendizaje (Bakki et al., 2020; Kali et al., 2015; Torres et al., 2014). Un aspecto en común que está presente en aquellas investigaciones en las que no se contempló en lo absoluto la Ingeniería de Software, es que los mismos autores de las publicaciones, al concluir sus investigaciones, mencionan la necesidad de un proceso que guíe el desarrollo del software, a la vez que identifican la complejidad que supone el diseño y creación de una plataforma que cumpla con todos los aspectos necesarios para apoyar en el proceso de enseñanza – aprendizaje (Bedregal-Alpaca et al., 2019; Gašević et al., 2017).

Esto último, de igual manera, se encuentra en las investigaciones que consideraron algunos principios, conceptos, procesos y metodologías de la Ingeniería de Software, donde incluso con conocimiento de causa, el producto desarrollado por los autores carece de las características esenciales (tanto funcionales como no funcionales) que todo producto de software debería poseer (Arvanitou et al., 2016). Es entendible la falta de pericia de algunos autores respecto a la labor de desarrollo, dada su formación profesional, la cual es prácticamente opuesta a una disciplina del área de las tecnologías de la información.

Entonces, como respuesta a la pregunta rectora de la presente revisión sistemática, ¿cómo se han considerado los principios de Ingeniería de Software en el diseño y desarrollo de plataformas de tecnología educativa?, es posible decir que los principios de la Ingeniería de Software (cuando son considerados) se valoran como elementos fundamentales del proceso de desarrollo de tecnología educativa aunque rara vez son incluidos en etapas de diseño, lo cual tiene un impacto directo sobre el resultado final, mismo que puede carecer de ciertas cualidades indispensables en el software.

Resultaría complejo pensar en algún modelo que contemplara y unificara tanto el sustento pedagógico, así como los principios del proceso de diseño y desarrollo de software. Sin embargo, esta unión podría darse en dos puntos de coincidencia que ambas disciplinas poseen, los patrones de diseño instruccional (que podrían sustentar el diseño y desarrollo de cursos en línea) y los patrones de diseño de software (que proveerían de estructura y formalidad al proceso de desarrollo).

## 5. ENLACES

Documentación de la revisión sistemática (términos de búsqueda, clasificación de los artículos, listado de referencias): <https://bit.ly/3u1gYzS>

Metadatos de la bibliografía consultada (archivo para importar en gestores de referencias): <https://bit.ly/3eZqYWr>

## 6. REFERENCIAS BIBLIOGRÁFICAS

- Adnan, N. H., & Ritzhaupt, A. D. (2018). Software Engineering Design Principles Applied to Instructional Design: What can we Learn from our Sister Discipline? *TechTrends*, 62(1), 77–94. <https://doi.org/10.1007/s11528-017-0238-5>
- Aeiad, E., & Meziane, F. (2019). An adaptable and personalised E-learning system applied to computer science Programmes design. *Education and Information Technologies*, 24(2), 1485–1509. <https://doi.org/10.1007/s10639-018-9836-x>
- Aguña, A. G. (2012). Patrones en aprendizaje: Concepto, aplicación y diseño de un patrón. *RED. Revista de Educación a Distancia*, 31, 1–19.
- Alharthi, A. D., Spichkova, M., & Hamilton, M. (2019). Sustainability requirements for eLearning systems: a systematic literature review and analysis. *Requirements Engineering*, 24(4), 523–543. <https://doi.org/10.1007/s00766-018-0299-9>
- Alzahrani, A. I., Al-Samarraie, H., Eldenfria, A., & Alalwan, N. (2018). A DEMATEL method in identifying design requirements for mobile environments: students' perspectives. *Journal of Computing in Higher Education*, 30(3), 466–488. <https://doi.org/10.1007/s12528-018-9176-2>
- Area-Moreira, M., Hernández-Rivero, V., & Sosa-Alonso, J. J. (2016). Modelos de integración didáctica de las TIC en el aula. *Comunicar*, 24(47), 79–87. <https://doi.org/10.3916/C47-2016-08>
- Arimoto, M. M., Barroca, L., & Barbosa, E. F. (2016). AM-OER: An agile method for the development of open educational resources. *Informatics in Education*, 15(2), 205–233. <https://doi.org/10.15388/infedu.2016.11>
- Arvanitou, E.-M., Ampatzoglou, A., Chatzigeorgiou, A., & Avgeriou, P. (2016). Software metrics fluctuation: a property for assisting the metric selection process. *Information and Software Technology*, 72, 110–124. <https://doi.org/10.1016/J.INFSOF.2015.12.010>
- Bakki, A., Oubahssi, L., George, S., & Cherkaoui, C. (2020). A Model and Tool to Support Pedagogical Scenario Building for Connectivist MOOC. *Technology, Knowledge and Learning*. <https://doi.org/10.1007/s10758-020-09444-8>
- Bedregal-Alpaca, N., Cornejo-Aparicio, V., Tupacyupanqui-Jaén, D., & Flores-Silva, S. (2019).

- Evaluación de la percepción estudiantil en relación al uso de la plataforma Moodle desde la perspectiva del TAM. *Ingeniare: Revista Chilena de Ingeniería*, 27(4), 707–718. <https://doi.org/10.4067/S0718-33052019000400707>
- Bennett, S., Bishop, A., Dalgarno, B., Waycott, J., & Kennedy, G. (2012). Implementing Web 2.0 technologies in higher education: A collective case study. *Computers and Education*, 59(2), 524–534. <https://doi.org/10.1016/j.compedu.2011.12.022>
- Bettio, R. W. de, Pereira, D. A., Martins, R. X., & Heimfarth, T. (2013). The Experience of Using the Scrum Process in the Production of Learning Objects for Blended Learning. *Informatics in Education*, 12(1), 29–41. <https://doi.org/10.15388/infedu.2013.03>
- Butler, A. E., Copnell, B., & Hall, H. (2018). The development of theoretical sampling in practice. *Collegian*, 25(5), 561–566. <https://doi.org/10.1016/j.colegn.2018.01.002>
- Campo, M., Amandi, A., & Biset, J. C. (2020). A software architecture perspective about Moodle flexibility for supporting empirical research of teaching theories. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-020-10291-4>
- Chen, W., Cheng, H. Y., & Bradley, E. (2017). Improving Online Teaching in a Required Geriatrics Clerkship Using Heuristic Evaluation. *Medical Science Educator*, 27(4), 871–875. <https://doi.org/10.1007/s40670-017-0437-x>
- Chimalakonda, S., & Nori, K. (2018). *A Patterns Based Approach for Design of Educational Technologies*. <http://arxiv.org/abs/1802.02663>
- Chimalakonda, S., & Nori, K. V. (2020). A family of software product lines in educational technologies. *Computing*, 102(8), 1765–1792. <https://doi.org/10.1007/s00607-019-00772-x>
- Chimalakonda, S., & Nori, K. V. (2012). A software engineering perspective for accelerating educational technologies. *Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies, ICALT 2012*, 754–755. <https://doi.org/10.1109/ICALT.2012.214>
- Chimalakonda, S., & Nori, K. V. (2013). What makes it hard to apply software product lines to educational technologies? *2013 4th International Workshop on Product Line Approaches in Software Engineering, PLEASE 2013 - Proceedings*, 17–20. <https://doi.org/10.1109/PLEASE.2013.6608657>
- Chounta, I. A., Manske, S., & Hoppe, H. U. (2017). “From Making to Learning”: introducing Dev Camps as an educational paradigm for Re-inventing Problem-based Learning. *International Journal of Educational Technology in Higher Education*, 14(1). <https://doi.org/10.1186/s41239-017-0061-2>
- Coomans, S., & Lacerda, G. S. (2015). PETESE, a Pedagogical Ergonomic Tool for Educational Software Evaluation. *Procedia Manufacturing*, 3, 5881–5888. <https://doi.org/10.1016/j.promfg.2015.07.895>
- Coutinho, E., & Bezerra, C. (2020). A study on dynamic aspects variability in the SOLAR educational software ecosystem. *Journal of the Brazilian Computer Society*, 26(1). <https://doi.org/10.1186/s13173-020-00103-5>
- Crescenzi-Lanna, L., & Grané-Oró, M. (2016). Análisis del diseño interactivo de las mejores apps educativas para niños de cero a ocho años. *Comunicar*, 24(46), 77–85. <https://doi.org/10.3916/C46-2016-08>
- Crowe, D., LaPierre, M., & Kebritchi, M. (2017). Knowledge Based Artificial Augmentation Intelligence Technology: Next Step in Academic Instructional Tools for Distance Learning. *TechTrends*, 61(5), 494–506. <https://doi.org/10.1007/s11528-017-0210-4>
- Damşa, C., Nerland, M., & Andreadakis, Z. E. (2019). An ecological perspective on learner-constructed learning spaces. *British Journal of Educational Technology*, 50(5), 2075–2089. <https://doi.org/10.1111/bjet.12855>
- Dehcordi, A. M., & Alavi, S. A. (2019). Structural analysis of Iranian educational technologies. *Journal of Global Entrepreneurship Research*, 9(1). <https://doi.org/10.1186/s40497-018->

[0130-6](#)

- Devedžić, V., & Jovanović, J. (2015). Developing Open Badges: A comprehensive approach. *Educational Technology Research and Development*, 63(4), 603–620. <https://doi.org/10.1007/s11423-015-9388-3>
- Dodero, J. M., Garcia-Peñalvo, F. J., González, C., Moreno-Ger, P., Redondo, M. A., Sarasa, A., & Sierra, J. L. (2012). Points of view on software engineering for eLearning (panel session). *2012 International Symposium on Computers in Education, SIIE 2012, May 2014*. <https://bit.ly/34aaepf>
- Douglas, I. (2001). Instructional design based on reusable learning objects: Applying lessons of object-oriented software engineering to learning systems design. *Proceedings - Frontiers in Education Conference*, 3. <https://doi.org/10.1109/fie.2001.963968>
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10), 833–859. <https://doi.org/10.1016/J.INFSOF.2008.01.006>
- Ertmer, P. A. (1999). Addressing first- and second-order barriers to change: Strategies for technology integration. *Educational Technology Research and Development*, 47(4), 47–61. <https://doi.org/10.1007/BF02299597>
- Flores-Fuentes, G., & Juárez-Ruiz, E. de L. (2017). Aprendizaje basado en proyectos para el desarrollo de competencias matemáticas en bachillerato. *Revista Electrónica de Investigación Educativa*, 19(3), 71–91. <https://doi.org/10.24320/redie.2017.19.3.721>
- Frimpon, M. F. (2012). A re-structuring of the critical success factors for e-learning deployment. *American International Journal of Contemporary Research*, 2(3), 115–127. <https://bit.ly/3bFTRoe>
- García, I., Pacheco, C., & García, W. (2009). A cooperative application to improve the educational software design using re-usable processes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5738 LNCS, 93–100. [https://doi.org/10.1007/978-3-642-04265-2\\_13](https://doi.org/10.1007/978-3-642-04265-2_13)
- Gašević, D., Mirriahi, N., Dawson, S., & Joksimović, S. (2017). Effects of instructional conditions and experience on the adoption of a learning tool. *Computers in Human Behavior*, 67, 207–220. <https://doi.org/10.1016/j.chb.2016.10.026>
- Gisbert, J. P., & Bonfill, X. (2004). ¿Cómo realizar, evaluar y utilizar revisiones sistemáticas y metaanálisis? *Gastroenterología y Hepatología*, 27(3), 129–149. <https://doi.org/10.1157/13058397>
- Golitsyna, I. (2015). Application of Web Services in Teaching of IT-Discipline. *Procedia - Social and Behavioral Sciences*, 214, 578–585. <https://doi.org/10.1016/j.sbspro.2015.11.763>
- González-González, C. S., Toledo-Delgado, P., & Muñoz-Cruz, V. (2015). Agile human centered methodologies to develop educational software. *DYNA (Colombia)*, 82(193), 187–194. <https://doi.org/10.15446/dyna.v82n193.53495>
- González-Martínez, J. A., Bote-Lorenzo, M. L., Gómez-Sánchez, E., & Cano-Parra, R. (2015). Cloud computing and education: A state-of-the-art survey. *Computers and Education*, 80, 132–151. <https://doi.org/10.1016/j.compedu.2014.08.017>
- Jabangwe, R., Edison, H., & Duc, A. N. (2018). Software engineering process models for mobile app development: A systematic literature review. *Journal of Systems and Software*, 145, 98–111. <https://doi.org/10.1016/J.JSS.2018.08.028>
- Kale, U., Yuan, J., & Roy, A. (2020). To design or to integrate? Instructional design versus technology integration in developing learning interventions. *Educational Technology Research and Development*. <https://doi.org/10.1007/s11423-020-09771-8>
- Kali, Y., McKenney, S., & Sagy, O. (2015). Teachers as designers of technology enhanced learning. *Instructional Science*, 43(2), 173–179. <https://doi.org/10.1007/s11251-014-9343-4>

- Kohen-Vacs, D., Milrad, M., Ronen, M., & Jansen, M. (2016). Evaluation of enhanced educational experiences using interactive videos and web technologies: pedagogical and architectural considerations. *Smart Learning Environments*, 3(1). <https://doi.org/10.1186/s40561-016-0029-z>
- Kugley, S., Wade, A., Thomas, J., Mahood, Q., Jørgensen, A. K., Hammerstrøm, K., & Sathe, N. (2017). Searching for studies: a guide to information retrieval for Campbell systematic reviews. En A. Aloe (Ed.), *Campbell Methods Series: Method Guide 1* (1.1). The Campbell Collaboration. <https://doi.org/10.4073/cm.2016.1>
- Leary, H., & Walker, A. (2018). Meta-Analysis and Meta-Synthesis Methodologies: Rigorously Piecing Together Research. *TechTrends*, 62(5), 525–534. <https://doi.org/10.1007/s11528-018-0312-7>
- Marcolino, A. S., & Barbosa, E. F. (2017). Towards a Software Product Line Architecture to Build M-learning Applications for the Teaching of Programming. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, 6264–6273. <https://doi.org/10.24251/hicss.2017.758>
- Mardis, M. A., Ma, J., Jones, F. R., Ambavarapu, C. R., Kelleher, H. M., Spears, L. I., & McClure, C. R. (2018). Assessing alignment between information technology educational opportunities, professional requirements, and industry demands. *Education and Information Technologies*, 23(4), 1547–1584. <https://doi.org/10.1007/s10639-017-9678-y>
- Martin, S., Diaz, G., Plaza, I., Ruiz, E., Castro, M., & Peire, J. (2011). State of the art of frameworks and middleware for facilitating mobile and ubiquitous learning development. *Journal of Systems and Software*, 84(11), 1883–1891. <https://doi.org/10.1016/j.jss.2011.06.042>
- Masson, P., & Udas, K. (2009). An agile approach to managing open educational resources. *On the Horizon*, 17(3), 256–266. <https://doi.org/10.1108/10748120910993286>
- McEneaney, J. E. (2016). Simulation-Based Evaluation of Learning Sequences for Instructional Technologies. *Instructional Science*, 44(1), 87–106. <https://doi.org/10.1007/s11251-016-9369-x>
- Meza Cano, J. M., Morales Ruiz, M. E., & Flores Macias, R. D. C. (2016). Variables individuales relacionadas con la instrucción en el uso de entornos personales de aprendizaje. *Educación*, 25(48), 87–106. <https://doi.org/10.18800/educacion.201601.005>
- Muyinda, P. B. (2007). MLearning: pedagogical, technical and organisational hypes and realities. *Campus-Wide Information Systems*, 24(2), 97–104. <https://doi.org/10.1108/10650740710742709>
- Nori, K. V., Reddy, Y. R., & Chimalakonda, S. (2014). Challenges for software engineering in educational technologies. *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, 267–272. <https://doi.org/10.1109/IC3I.2014.7019725>
- Osuna-Acedo, S., Marta-Lazo, C., & Frau-Meigs, D. (2018). De sMOOC a tMOOC, el aprendizaje hacia la transferencia profesional: El proyecto europeo ECO. *Comunicar*, 26(55), 105–114. <https://doi.org/10.3916/C55-2018-10>
- Oyelere, S. S., & Suhonen, J. (2016). Design and implementation of MobileEdu M-learning application for computing education in Nigeria: A design research approach. *Proceedings - 2016 International Conference on Learning and Teaching in Computing and Engineering, LaTiCE 2016*, 27–31. <https://doi.org/10.1109/LaTiCE.2016.3>
- Oyelere, S. S., Suhonen, J., Wajiga, G. M., & Sutinen, E. (2018). Design, development, and evaluation of a mobile learning application for computing education. *Education and Information Technologies*, 23(1), 467–495. <https://doi.org/10.1007/s10639-017-9613-2>
- Pástor, D., Jiménez, J., Arcos, G., Romero, M., & Urquiza, L. (2018). Patrones de diseño para la construcción de cursos on-line en un entorno virtual de aprendizaje. *Ingeniare. Revista Chilena de Ingeniería*, 26(1), 157–171. <http://dx.doi.org/10.4067/S0718->

[33052018000100157](https://doi.org/10.1109/EAAEIE.2009.5335484)

- Plaza, I., Marcuello, J. J., Igual, R., & Arcega, F. (2009). Proposal of a quality model for educational software. *20th EAAEIE Annual Conference, EAAEIE 2009 - Formal Proceedings*. <https://doi.org/10.1109/EAAEIE.2009.5335484>
- Richardson, J. C., Brush, T., Ottenbreit-Leftwich, A., Karlin, M., Leary, H., Shelton, B. E., Lowell, V., Exter, M. E., Strycker, J., & Shin, S. (2020). Innovations in Instructional Design and Technology Programs: a View from PIDT 2018. *TechTrends*, 64(3), 432–438. <https://doi.org/10.1007/s11528-019-00445-8>
- Ritzhaupt, A. D., & Kumar, S. (2015). Knowledge and Skills Needed by Instructional Designers in Higher Education. *Performance Improvement Quarterly*, 28(3), 51–69. <https://doi.org/10.1002/piq.21196>
- Rook, M., Choi, K., & McDonald, S. (2015). Learning Theory Expertise in the Design of Learning Spaces: Who Needs a Seat at the Table? *Journal of Learning Spaces*, 4(1), 17–29.
- Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., & Suthers, D. (1999). Developing educational software components. *Computer*, 32(9), 50–58. <https://doi.org/10.1109/2.789751>
- Salinas, J. I., & Marín, V. I. J. (2019). Metasíntesis cualitativa sobre colaboración científica e identidad digital académica en redes sociales. *RIED. Revista Iberoamericana de Educación a Distancia*, 22(2), 97. <https://doi.org/10.5944/ried.22.2.23238>
- Sengupta, S., Chaki, N., & Dasgupta, R. (2009). Learners' Quanta based Design of a Learning Management System. *International Journal of Education and Information Technologies*, 3(1), 67–74. <https://bit.ly/3wj62zc>
- Sengupta, S., & Dasgupta, R. (2017a). Architectural design of a LMS with LTSA-conformance. *Education and Information Technologies*, 22(1), 271–296. <https://doi.org/10.1007/s10639-015-9443-z>
- Sengupta, S., & Dasgupta, R. (2017b). LTSA conformance testing to architectural design of LMS using ontology. *Education and Information Technologies*, 22(6), 3017–3035. <https://doi.org/10.1007/s10639-016-9569-7>
- Setirek, A. C., & Tanrikulu, Z. (2015). Significant Developmental Factors that can Affect the Sustainability of Mobile Learning. *Procedia - Social and Behavioral Sciences*, 191, 2089–2096. <https://doi.org/10.1016/j.sbspro.2015.04.336>
- Squires, D., & Preece, J. (1999). Predicting quality in educational software: Evaluating for learning, usability and the synergy between them. *Interacting with Computers*, 11(5), 467–483. [https://doi.org/10.1016/S0953-5438\(98\)00063-0](https://doi.org/10.1016/S0953-5438(98)00063-0)
- Sun, P. C., Tsai, R. J., Finger, G., Chen, Y. Y., & Yeh, D. (2008). What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction. *Computers and Education*, 50(4), 1183–1202. <https://doi.org/10.1016/j.compedu.2006.11.007>
- Tabares Morales, V., Duque Méndez, N. D., & Carranza, Ovalle, D. A. (2017). Modelo por capas para evaluación de la calidad de Objetos de Aprendizaje en repositorios. *Revista Electrónica de Investigación Educativa*, 19(3), 33–48. <https://doi.org/10.24320/redie.2017.19.3.1128>
- Thomas, J., & Harden, A. (2008). Methods for the thematic synthesis of qualitative research in systematic reviews. *BMC Medical Research Methodology*, 8. <https://doi.org/10.1186/1471-2288-8-45>
- Torres, J., Resendiz, J., Aedo, I., & Dodero, J. M. (2014). A model-driven development approach for learning design using the LPCEL Editor. *Journal of King Saud University - Computer and Information Sciences*, 26(1), 17–27. <https://doi.org/10.1016/j.jksuci.2013.10.004>
- Vega López, N. A., Perales-Escudero, M. D., Correa Gutiérrez, S., Murrieta Loyo, G., & Reyes

- Cruz, M. del R. (2019). Validación de la escala de medida de prácticas educativas de la lectura comparativa. *Revista mexicana de investigación educativa*, 24(83), 1077–1107. <https://bit.ly/3hzbCcM>
- Yeoman, P., & Ashmore, N. (2018). Moving from pedagogical challenge to ergonomic challenge: Translating epistemology into the built environment for learning. *Australasian Journal of Educational Technology*, 34(6), 1–16. <https://doi.org/10.14742/ajet.4502>
- Yulianto, B., Heriyanni, E., Dewi, L. C., & Adinugroho, T. Y. (2015). Architecture and Implementation of Instant Messaging in Educational Institution. *Procedia Computer Science*, 59, 5–13. <https://doi.org/10.1016/j.procs.2015.07.331>
- Zadahmad, M., & Yousefzadehfard, P. (2016). Agile development of various computational power adaptive web-based mobile-learning software using mobile cloud computing. *International Journal of Web-Based Learning and Teaching Technologies*, 11(2), 61–72. <https://doi.org/10.4018/IJWLTT.2016040104>
- Zhang, S., & Dorn, B. (2012). Accelerating Software Development through Agile Practices - A Case Study of a Small-scale, Time-intensive Web Development Project at a College-level IT Competition. *Journal of Information Technology Education: Innovations in Practice*, 11, 25–37. <https://doi.org/10.28945/1545>

---

## INFORMACIÓN SOBRE LOS AUTORES

### Mauricio Arturo Ibarra-Corona

Facultad de Informática de la Universidad Autónoma de Querétaro

Ingeniero de Software egresado de la Facultad de Informática de la Universidad Autónoma de Querétaro y con el grado de Maestro en Sistemas de Información por parte de la misma universidad. Docente e investigador en las áreas de desarrollo web, ingeniería de software y programación. Actualmente, cursa el Doctorado en Innovación en Tecnología Educativa, siguiendo líneas de investigación relativas a la ingeniería de software, desarrollo de web, arquitectura de software y desarrollo de tecnología educativa.

### Alexandro Escudero-Nahón

Facultad de Informática de la Universidad Autónoma de Querétaro

Doctor en Educación por la Universidad de Barcelona. Pertenece al Sistema Nacional de Investigadores en el Nivel 1 del Consejo Nacional de Ciencia y Tecnología de México (CONACyT). Es profesor investigador de tiempo completo en la Facultad de Informática de la Universidad Autónoma de Querétaro. Coordina el Doctorado en Tecnología Educativa en esta universidad. Dirige el proyecto *Transdigital*, que es una iniciativa ciudadana para la difusión de la ciencia con cuatro líneas de trabajo: revista científica *Transdigital*, Congreso Virtual *Transdigital*, Divulgación *Transdigital* y Editorial Electrónica *Transdigital*. Coordina el Comité de Investigación y Posgrados en Tecnología Educativa de la Red LaTE México, que es una Red Temática CONACyT. Forma parte del Cuerpo Académico Consolidado “Innovación Educativa y Tecnología” de la Secretaría de Educación Pública de México (SEP).



Los textos publicados en esta revista están sujetos a una licencia de Reconocimiento 4.0 España de Creative Commons. Puede copiarlos, distribuirlos, comunicarlos públicamente y hacer obras derivadas siempre que reconozca los créditos de las obras (autoría, nombre de la revista, institución editora) de la manera especificada por los autores o por la revista. La licencia completa se puede consultar en: [Licencia Creative Commons Atribución-NoComercial-Compartir por igual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).