

Universidad Autónoma de Querétaro
Facultad de Informática

Metodología para el desarrollo de plataformas de tecnología educativa con
base en patrones arquitectónicos y de diseño de software

Tesis

Que como parte de los requisitos
para obtener el Grado de
Doctor en Innovación en Tecnología Educativa

Presenta

Mauricio Arturo Ibarra Corona

Dirigido por:

Dr. Alexandro Escudero Nahón

Querétaro, Qro. a 06 de febrero de 2023



Dirección General de Bibliotecas y Servicios Digitales
de Información



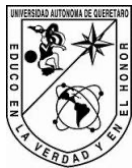
Metodología para el desarrollo de plataformas de
tecnología educativa con base en patrones
arquitectónicos y de diseño de software

por

Mauricio Arturo Ibarra Corona

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Clave RI: IFDCC-228344



Universidad Autónoma de Querétaro
Facultad de Informática
Doctorado en Innovación en Tecnología Educativa

Metodología para el desarrollo de plataformas de tecnología educativa con base en
patrones arquitectónicos y de diseño de software

Tesis

Que como parte de los requisitos para obtener el Grado
Doctor en Innovación en Tecnología Educativa

Presenta

Mauricio Arturo Ibarra Corona

Dirigido por:

Dr. Alexandro Escudero Nahón

Dr. Alexandro Escudero Nahón
Presidente

Dra. Sandra Luz Canchola Magdaleno
Secretario

Dr. Alberto Lara Guevara
Vocal

Dra. Elvia Garduño Teliz
Suplente

Dra. Ma. Teresa García Ramírez
Suplente

Centro Universitario, Querétaro, Qro.
Febrero, 2023
México

Dedicatorias

A mi abuelito, José Corona Monge, quien siempre me alentó a ser mejor y me daba consejos llenos de sabiduría que siempre llevaré conmigo. Donde quiera que estés, sé que estás en mi corazón.

A mi mamá, Rosario Esther Corona Orozco, sin la cual no habría llegado hasta este momento y que ha estado conmigo toda la vida, ojalá nunca me faltes, mamá.

A mi hermano, Diego Octavio Ibarra Corona, que desde pequeño me cuidó, guio y ha sido siempre un modelo a seguir.

A mi novia, Paulina Jiménez Peña, con la que discutí muchas de las ideas aquí vertidas y que desde que estaba en la carrera me impulsa a seguir adelante.

A mi hermano de otra madre, José Alejandro Vargas Díaz, quien siempre hacía las preguntas importantes y me ha brindado su apoyo desde que lo conocí.

Agradecimientos

A mi director de tesis, Dr. Alexandro Escudero Nahón, quién conocí durante la licenciatura y que, desde entonces me enseñó las bases de cualquier tipo de investigación. Muchas gracias por todo su apoyo, orientación y consejos.

A mis sinodales, Dra. Sandra Luz Canchola Magdaleno, Dr. Alberto Lara Guevara, Dra. Elvia Garduño Teliz y Dra. Ma. Teresa García Ramírez. Quienes con sus invaluable aportaciones me orientaron y guiaron en el desarrollo de este proyecto de investigación, muchas gracias por sus siempre acertados comentarios y cuestionamientos. Fueron una parte fundamental de mi formación y, por lo mismo, les estaré siempre agradecido.

A mi compañero de guerra, con el que cursé maestría y doctorado al mismo tiempo, aunque en diferentes planes, M.C.C. Martín Muñoz Mandujano, muchas gracias por todo tu apoyo y por ser siempre un gran compañero, un gran amigo y alguien quien sin duda alguna puedo llamar hermano.

A la Dra. Gabriela Xicotécatl Ramírez, directora de la Facultad de Informática, que siempre ha sido un ejemplo sobre el buen actuar profesional, personal y académico. Gracias por todos sus consejos desde que era mi tutora en licenciatura, a la fecha los tengo bien presentes y los uso en mi quehacer personal y profesional. A la Dra. Rosa María Romero González, quien desde la maestría me orientó y aconsejó como investigador. Atesoro mucho todas sus técnicas, métodos y consejos, ya que me permitieron entender todo lo que significaba realizar una investigación.

Finalmente, a mi familia, amigos, amigas y al Centro de Desarrollo de la Facultad de Informática, los cuáles siempre han estado presentes en los buenos y malos momentos y que siempre son una gran fuente de inspiración.

Índice

Resumen.....	1
Abstract.....	2
1. Introducción.....	3
1.1. Justificación.....	3
1.2. Descripción del problema.....	4
2. Antecedentes.....	11
2.1. Ingeniería de Software.....	11
2.2. Diseño Instruccional.....	14
2.3. Proceso de enseñanza-aprendizaje.....	15
3. Marco teórico.....	18
3.1. Tecnología educativa.....	18
3.2. Diseño instruccional.....	21
3.3. Proceso de desarrollo de software.....	23
3.4. Arquitectura de software.....	27
3.5. Patrones de diseño de software.....	28
3.6. Diseño de software educativo.....	33
4. Planteamiento teórico.....	35
4.1. Hipótesis.....	35
4.2. Preguntas de investigación.....	35
4.3. Objetivos.....	35
4.3.1. Objetivo general.....	35
4.3.2. Objetivos específicos.....	35
5. Metodología.....	37
5.1. Análisis conceptual de la temática de investigación.....	37
5.2. Desarrollo de la propuesta metodológica.....	39
5.3. Implementación de la propuesta metodológica.....	39
5.4. Validación de la propuesta.....	39
5.5. Producción del material suplementario.....	40
6. Propuesta.....	41

6.1. Modelo unificador	41
6.2. Metodología propuesta	47
7. Resultados	55
8. Discusión.....	66
9. Conclusión.....	69
Referencias bibliográficas	72
Anexos	87
Anexo 1. Encuesta para la definición del problema de la investigación	87
Anexo 2. Entrevista para evaluación de la metodología desarrollada.....	94

Índice de tablas

Tabla 1.1. Temáticas tratadas por los autores	4
Tabla 2.1. Contexto histórico del diseño instruccional	14
Tabla 3.1. Enfoques de investigación sobre tecnología educativa	19
Tabla 3.2. Modalidades afines a la tecnología educativa	20
Tabla 3.3. Modelos de diseño instruccional	21
Tabla 3.4. Descripción de metodologías ágiles	25
Tabla 3.5. Patrones arquitectónicos de software	28
Tabla 3.6. Composición de los patrones de diseño	29
Tabla 3.7. Patrones de diseño creacionales	30
Tabla 3.8. Patrones de diseño estructurales.....	31
Tabla 3.9. Patrones de diseño conductuales	31
Tabla 3.10. Aspectos necesarios en el diseño de software educativo.....	34
Tabla 6.1. Clasificación categórica de los conceptos según su área.....	42
Tabla 6.2. Definición de los patrones estructurales y funcionales	44
Tabla 6.3. Análisis comparativo de las metodologías	48
Tabla 6.4. Descripción de las etapas y procesos de la metodología propuesta	51
Tabla 7.1. Categorías temáticas resultantes del análisis cualitativo de las entrevistas	55
Tabla 7.2. PETESE aplicado a la plataforma desarrollada	61

Índice de figuras

Figura 2.1. Modelo en cascada original	12
Figura 2.2. Metodología RAD.....	13
Figura 2.3. Componentes del proceso de enseñanza-aprendizaje.....	16
Figura 3.1. Metodología RUP.....	24
Figura 3.2. Comparativa de modelos de diseño instruccional y de desarrollo de software.....	26
Figura 5.1. Metodología de investigación	37
Figura 6.1. Delimitación conceptual de la tecnología educativa y la ingeniería de software	41
Figura 6.2. Modelo unificador.....	43
Figura 6.3. Metodología propuesta para el desarrollo de plataformas educativas	50
Figura 6.4. Metodología propuesta sintetizada	54
Figura 7.1. Frecuencia relativa de las categorías temáticas.....	56
Figura 7.2. Codificación axial de las categorías temáticas	57
Figura 7.3. Interfaces de la plataforma desarrollada con base en la metodología	60
Figura 7.4. Metodología propuesta (post-retroalimentación)	64
Figura 7.5. Segunda versión de la plataforma desarrollada.	65

Resumen

La pandemia suscitada en el año 2020 se trató de un punto de inflexión que ha modificado por completo la manera en la que se imparten clases. El cambio tan súbito de paradigma al que se vio sometida la educación permitió develar carencias presentes en la infraestructura, funcionalidad y usabilidad en las herramientas y plataformas de tecnología educativa, mismas que se derivan de la evidente falta de un proceso de desarrollo de software específico para plataformas de tecnología educativa que contemple los elementos teóricos del diseño instruccional. Derivado de lo anterior, se estableció como objetivo diseñar una metodología de desarrollo de software para la creación de plataformas de tecnología educativa haciendo uso patrones arquitectónicos y de diseño de software que facilite la aplicación de modelos de diseño instruccional en dichas plataformas. Se utilizó la Investigación Basada en Diseño en cinco etapas para el desarrollo de un modelo capaz de unificar la Ingeniería de Software y el Diseño Instruccional, a partir del cual fue posible diseñar la metodología propuesta en este proyecto de investigación, misma que fue probada y verificada de forma cualitativa y cuantitativa mediante su utilización en el diseño de interfaces de software para una plataforma de tecnología educativa. Se concluye que no solo es posible utilizar conceptos de disciplinas con campos semánticos diferentes, sino que es posible utilizarlos de forma complementaria y sinérgica. Además, se encontró que la metodología puede funcionar incluso como referencia para seleccionar herramientas y plataformas de desarrollo existentes, sin la necesidad de iniciar un proceso de desarrollo nuevo. Además, a partir de la metodología se plantea la posibilidad de generar un instrumento capaz de sugerir qué plataforma o herramienta de tecnología educativa es más adecuada en un determinado escenario para un docente.

Palabras clave: *Diseño Instruccional, Ingeniería de Software, Patrones de Diseño, Software Educativo, Tecnología Educativa.*

Abstract

The pandemic that arose in 2020 was a turning point that has completely changed the way classes are taught. The sudden change of paradigm to which education was subjected allowed revealing shortcomings in the infrastructure, functionality and usability of educational technology tools and platforms, which derive from the evident lack of a specific software development process for educational technology platforms that contemplate the theoretical elements of instructional design. Derived from the above, the objective was to design a software development methodology for the creation of educational technology platforms using architectural and software design patterns that facilitate the application of instructional design models on said platforms. Research Based on Design was used in five stages for the development of a model capable of unifying Software Engineering and Instructional Design, from which it was possible to design the methodology proposed in this research project, which was tested and verified. qualitatively and quantitatively through its use in the design of software interfaces for an educational technology platform. It is concluded that not only is it possible to use concepts from disciplines with different semantic fields, but it is also possible to use them in a complementary and synergistic way. In addition, it was found that the methodology can even work as a reference to select existing development tools and platforms, without the need to start a new development process, in addition, based on the methodology, the possibility of generating an instrument capable of suggesting which educational technology platform or tool is most appropriate in each scenario for a teacher.

Keywords: *Design Patterns, Educational Software, Educational Technology, Instructional Design, Software Engineering.*

1. Introducción

1.1. Justificación

La pandemia suscitada en el año 2020 se trató de un punto de inflexión que ha modificado por completo la manera en la que se imparten clases. Ante dicha situación, las plataformas de tecnología educativa han tomado especial relevancia e importancia en la educación de estudiantes de todo el mundo (Viner et al., 2020). El cambio tan súbito de paradigma al que se vio sometida la educación tradicional y presencial permitió develar carencias presentes en la infraestructura, funcionalidad y usabilidad de múltiples herramientas de tecnología educativa. Fue evidente que muchas de estas soluciones no estaban preparadas ni diseñadas para convertirse en el único medio a través del cual se pudieran impartir clases (Bayham & Fenichel, 2020). Se volvió más importante que nunca que las plataformas estén diseñadas y adecuadas para funcionar con los modelos de diseño instruccional que el docente utilice.

Una metodología de desarrollo de software provee de estructura y permite la creación de un producto alineado con las necesidades del cliente y del usuario final. Imponen un proceso disciplinado, lo que busca convertirlo en una actividad eficiente, predecible y planificada (Delgado, 2008). Lo anterior es especialmente relevante cuando se considera que las plataformas de tecnología educativa son proyectos con necesidades que requieren un acercamiento especial, sobre todo, cuando se trata de su diseño y desarrollo (Bartolomé et al., 2018) y en donde el docente puede ser considerado como cliente y usuario final (en conjunto con los estudiantes).

Así como las metodologías resuelven un problema muy común, que es definir el camino a seguir al momento de diseñar y desarrollar un software, los patrones de diseño de software existen como soluciones a problemas y necesidades constantemente presentes en proyectos de software (Gamma et al., 1995), mientras que la arquitectura de software permite la representación adecuada de cualquier sistema desarrollado o por desarrollar.

En este sentido, los patrones de arquitectura y de diseño de software, implementados en una metodología de desarrollo de software que busque su concordancia con los patrones de diseño instruccional (Chimalakonda & Nori, 2018), lograrían la unión entre la teoría y la tecnología que menciona Lowyck (2014) y que beneficiaría, tanto a docentes como a alumnos, permitiendo el desarrollo de plataformas flexibles, de calidad y que se adapten a su forma de enseñanza, evitando así que los docentes se tengan que adaptar a la plataforma, lo cual mejoraría la percepción, flexibilidad, adaptabilidad, escalabilidad y usabilidad de las plataformas de tecnología educativa en ambientes educativos de nivel superior.

1.2. Descripción del problema

Para la identificación del problema se realizó una investigación bibliográfica referente al diseño y elaboración de plataformas de tecnología educativa. La investigación se realizó en las bases de datos científicas *Science Direct*, *Google Scholar*, *Springer* y *Research Gate*, se utilizó como término de búsqueda: “Diseño” AND “Tecnología Educativa” AND “Software”. Se revisaron en total 30 materiales bibliográficos publicados desde el 2002 hasta el 2019.

Tras realizar la revisión, se encontraron 21 temáticas que los diversos autores trataron al desarrollar artículos sobre tecnología educativa. En la Tabla 1.1 se muestran las temáticas y frecuencias con las que fueron tratadas en los materiales.

Tabla 1.1.

Temáticas tratadas por los autores

Temática	Frecuencia	Porcentaje
Tecnología educativa	26	96%
Diseño Instruccional	24	89%
Teoría de la enseñanza	19	70%
Teoría del aprendizaje	16	59%
Entornos digitales de aprendizaje	16	59%
e-Learning	15	56%

Temática	Frecuencia	Porcentaje
Tecnologías de la Información y Comunicación	15	56%
Apoyo del aprendizaje	13	48%
Adquisición del conocimiento	13	48%
Desarrollo de habilidades	12	44%
Technology Enhanced Learning	12	44%
Experiencia de aprendizaje	11	41%
Competencias digitales	10	37%
Calidad educativa	8	30%
Diseño Instruccional Digital	8	30%
Recursos Educativos Libres (OER)	6	22%
Procesamiento de la información	5	19%
Cursos Online Masivos y Abiertos (MOOC)	5	19%
Dispositivos móviles	3	11%
Diseño de software	3	11%
Diseño Centrado en el Usuario	1	4%

Una vez identificadas las temáticas y la frecuencia con las que fueron mencionadas en la literatura revisada, es necesario establecer el contexto en el cual se desarrollan, con la finalidad de definir concretamente la problemática que la presente investigación plantea.

Marín et al. (2018), Marín et al. (2017), Zawacki-Richter et al. (2017) y Sánchez et al. (2014) describen en sus trabajos las diferentes temáticas que se han tratado en revistas de tecnología educativa en la última década y media. Hew et al. (2019) y Ahmad & Nisa (2016), tratan en sus respectivos artículos el impacto que tiene el proceso de enseñanza-aprendizaje en la tecnología y como la tecnología ha modificado dicho proceso. Por otra parte, Hoogveld et al. (2002) y Bennett et al. (2017) hablan respecto a las prácticas y costumbres que tienen los docentes al

momento de diseñar sus cursos y materiales para clase. Siguiendo esa misma línea, Ertmer (2005) y Mishra (2002) plantean si el diseño instruccional que cada docente realiza representará una limitante en la transición hacia la tecnología educativa.

Según las investigaciones de McKnight et al. (2016), Spector (2017), Ipek y Ziatdinov (2018), Bennett et al. (2017) y Niederhauser et al. (2018), las implicaciones de la tecnología educativa sobre los procesos de enseñanza-aprendizaje han tenido un impacto directo en las investigaciones referentes a ambos temas. Esto, diversifica ambas áreas del conocimiento, ya que la tecnología educativa se puede entender como el espacio a través del cual se desarrolla dicho proceso (como en los MOOC) o únicamente como una herramienta de apoyo (tal es el caso de los Recursos Educativos Libres), lo que aumenta la complejidad del desarrollo de las investigaciones y de la aplicación de estas.

Henderson et al. (2017) y Englund et al. (2017), por su parte, tienen un acercamiento distinto al del resto de los autores, al realizar un análisis de la percepción de los estudiantes respecto a la tecnología educativa. En sus artículos, realizan un estudio de aquellos aspectos que funcionan y son efectivos y aquellos que no lo son, vinculándolo directamente con el diseño instruccional que cada docente elabora. A su vez, Lee et al. (2017), Antonenko et al. (2017) y Ausburn & Brown (2006), mencionan la necesidad de alinear las necesidades y capacidades de los docentes y de los alumnos utilizando tecnología educativa, ya que reconocen que existe una brecha de habilidades y competencias para su uso adecuado.

Derivado de lo anterior, Lowyck (2014), Berggren et al. (2005) y FitzGerald et al. (2018) mencionan que existe una evidente brecha entre la teoría de la tecnología educativa y la práctica de esta. Mencionan lo complicado que es replicar un modelo pedagógico en herramientas de tecnología educativa. Así pues, los diferentes autores coinciden que esta brecha puede deberse a la incompatibilidad de la tecnología educativa con los procesos de enseñanza-aprendizaje tradicionales, a la ausencia de uno o varios modelos de diseño instruccional por

parte del docente o simplemente por una falta de competencias y habilidades digitales.

Bajo la misma línea se encuentran Bartolomé et al. (2018), Koehler & Mishra (2005), Kahle (2008) y Nguyen & Bower (2018), quienes describen un problema que está presente en la inmensa mayoría de las herramientas y plataformas de tecnología educativa: la falta de un fundamento pedagógico tras el diseño y concepción de dichas herramientas. Es decir, resaltan la carencia de un diseño instruccional claro en las aplicaciones de tecnologías educativas, lo que resulta en una incompatibilidad entre lo planeado y diseñado por los docentes, los procesos de enseñanza-aprendizaje y las capacidades de la tecnología desarrollada. Lo anterior se ve reforzado por lo expuesto por Chimalakonda & Nori (2018), Kirschner (2015) y Strecker et al. (2018), quienes plantean la necesidad de incluir al docente en el proceso del diseño y desarrollo de software de tecnología educativa.

Realizado el análisis de la investigación bibliográfica, se pudieron identificar los siguientes conceptos como pertinentes y de interés para la presente investigación: diseño instruccional, diseño instruccional digital y diseño de software.

De las temáticas, diseño instruccional, es la más mencionada por los autores, con un 89% de coincidencias en los materiales revisados. En la mayoría de los casos, este concepto es abordado como una labor docente que consiste en la elaboración y planeación de la didáctica de un curso. Sin embargo, al revisar las otras dos temáticas de interés (diseño instruccional digital y diseño de software), es posible notar que únicamente el 30% y el 11% respectivamente de los autores mencionaron estos temas. Si bien se hace mención del diseño instruccional al momento de hablar sobre tecnología educativa, muy pocos diferencian entre planear una clase de forma tradicional y planear una con herramientas tecnológicas.

Es evidente la relación existente entre los problemas mencionados por los diferentes autores (poca efectividad percibida por los alumnos y mala percepción por parte de los docentes) y la falta de un sustento pedagógico en las aplicaciones

de tecnología educativa. El hecho de que tan pocos autores (únicamente el 11%) contemplen el diseño del software cuando se habla del diseño de tecnología educativa denota un claro problema en la teoría referente a la misma, en específico la falta de convergencia entre las propuestas de diseño instruccional con el diseño de software educativo. Sin embargo, en cualquier tema existe una brecha entre la teoría y la práctica, por lo que es necesario realizar la siguiente pregunta, ¿es posible apreciar este mismo problema en la práctica?

Para contestar lo anterior y lograr complementar lo encontrado en la investigación bibliográfica, se realizó una encuesta a 26 docentes de la Facultad de Informática de la Universidad Autónoma de Querétaro (FIF - UAQ). La encuesta constó de 31 preguntas, siete de las cuales eran demográficas y 24 que tenían como propósito medir diversos indicadores (definidos a partir de los resultados de la investigación bibliográfica), estas preguntas hicieron uso de la escala Likert para facilitar su posterior análisis y comprobación. El instrumento aplicado fue analizado haciendo uso de la plataforma SPSS de IBM, donde se obtuvo un Alfa de Cronbach de 0.845, lo que, según este índice de fiabilidad, significa que se trata de un instrumento con validez metodológica.

De los 26 docentes encuestados, los datos demográficos son los siguientes: nueve eran hombres (34.6%) y 17 eran mujeres (65.5%). Respecto a su edad, 30.8% tenía más de 35 años, 23.1% entre 24 y 27 años, 19.2% entre 28 y 31 años, 15.4% 23 años o menos y 11.5% entre 32 y 35 años. Cuando se les preguntó sobre la cantidad de años que han impartido clases, se obtuvo como resultado que el 42.3% ha impartido clases durante nueve años o más, 26.9% dos años o menos, 19.2% entre tres y cinco años y 11.5% entre seis y ocho años. De estos datos, es relevante mencionar que los demográficos no tuvieron un impacto significativo en los resultados obtenidos.

De igual manera, y con la finalidad de encontrar alguna relación entre el área de formación y los resultados obtenidos, se les preguntó a los encuestados el área de conocimiento en la cual recibieron su educación profesional. El 61.5% de los

encuestados contestó que se formaron en el Área de Tecnologías de la Información y Comunicación, 34.6% en Ciencias Sociales y 3.8% en Ciencias de la Salud. Con los resultados demográficos es posible determinar que el grupo de encuestados es bastante diverso en cuanto años de experiencia, edades y áreas del conocimiento.

Como primer indicador, se utilizó el conocimiento existente respecto al Diseño Instruccional (DI). Se encontró que el 96.2% de los encuestados realiza regularmente diseño instruccional para un curso mientras que solo el 3.8% lo hace de manera esporádica. Se les preguntó a los docentes si hacían uso de alguna plataforma de tecnología educativa, el 80.8% de los docentes contestó que regularmente hace uso de una para la impartición de sus clases, mientras que el 19.2% lo hace de forma ocasional, lo que representa un uso bastante elevado y constante de dicho tipo de herramienta.

El segundo indicador es referente al Diseño Instruccional Digital (DID). Cuando se le preguntó a los encuestados si habían tenido que realizar alguna modificación al diseño y planeación de su curso debido a la plataforma de tecnología educativa, el 50% tuvo una respuesta con tendencia negativa y el 50% con tendencia positiva, lo cual refleja hasta cierto punto la magnitud de la brecha entre la pedagogía y la tecnología que mencionan Bartolomé et al. (2018) y Lowyck (2014).

Para complementar la pregunta anterior se les preguntó si realizaban alguna planeación especial al momento de utilizar una plataforma, el 46.2% mencionó que hacían esto algunas veces, el 19.2% casi siempre y el 3.8% siempre lo hacía, mientras que el 11.5% mencionó que nunca lo hacía y el 19.2% casi nunca. Estos resultados demuestran en cierta medida la incompatibilidad del diseño instruccional realizado por los docentes y la plataforma de tecnología educativa, lo que puede ser debido a una mala elección de la plataforma o por la falta de sustento pedagógico de la misma.

Ahora bien, lo anterior determina, en gran medida, el éxito del siguiente indicador a medir, la implementación del diseño instruccional en plataformas de tecnología educativa. Con esto en mente, se preguntó si las plataformas de tecnología limitaban el tipo de enseñanza que se podía dar. Se obtuvieron resultados diametralmente opuestos, estando de acuerdo 42.3% de los docentes, en desacuerdo 34.6% e indiferentes el 23.1%.

Tras el análisis de los resultados obtenidos con el instrumento y contrastándolo con lo mencionado por los diversos autores vistos en la revisión bibliográfica, es posible llegar a las siguientes deducciones:

- Existe una brecha entre la pedagogía, la didáctica y el software educativo, tal como mencionan los autores y es evidente que los docentes tienen que adecuar sus cursos a lo que una plataforma les permita. Esto es derivado del perfil docente, mismo que no está orientado a la pedagogía.
- Las plataformas carecen de ciertas capacidades y características descritas por los autores y tienen un impacto directo en el proceso de enseñanza-aprendizaje. Entre las carencias, se pueden mencionar: la accesibilidad, la usabilidad, la escalabilidad, la flexibilidad, la portabilidad, el impulso a la motivación, los espacios para la exploración y la capacidad para la cooperación y colaboración, todas de vital importancia en todo proyecto de software educativo.
- En algunas instancias, el docente tiene que adaptar sus necesidades de enseñanza al software, en lugar de que el software se adapte a él, lo que representa un serio problema para cualquier tecnología y cuya solución es el correcto diseño arquitectónico en conjunto con políticas y estándares de calidad bien definidos (Opel & Rhodes, 2018).

Establecido lo anterior, es evidente la falta de un proceso de desarrollo de software específico para plataformas de tecnología educativa bien definido y que contemple los aspectos elementales de los modelos de diseño instruccional mencionados en la teoría.

2. Antecedentes

El desarrollo de plataformas de tecnología educativa, pese a tratarse de un tema relativamente nuevo y a estar íntimamente ligado a la inclusión de la tecnología en entornos educativos, ha sido motivo de múltiples investigaciones. Desde sus orígenes en la formación militar, hasta la actualidad donde se presenta como un mediador ante el nuevo paradigma de aprendizaje con conceptos consolidados como el *e-learning*. Las plataformas de tecnología educativa representan una evolución natural de la tecnología educativa, que en sus inicios buscaba hacer más eficientes los procesos de enseñanza-aprendizaje con el uso de recursos y medios tecnológicos (Area, 2009).

Las plataformas de tecnología educativa, sin embargo, consideran aspectos que van más allá de lo que la tecnología educativa por sí sola contempla. El desarrollo de estas plataformas se ve condicionado a la consideración y puntos de vista de tres disciplinas: ingeniería de software, diseño instruccional y el proceso de enseñanza-aprendizaje (Chimalakonda & Nori, 2018). Lo anterior, implica que la tecnología educativa es una ciencia de diseño desde su concepción y es la encargada de coordinar los diferentes conocimientos de las otras disciplinas para la creación de plataformas de tecnología educativa (Ahmad & Nisa, 2016).

2.1. Ingeniería de Software

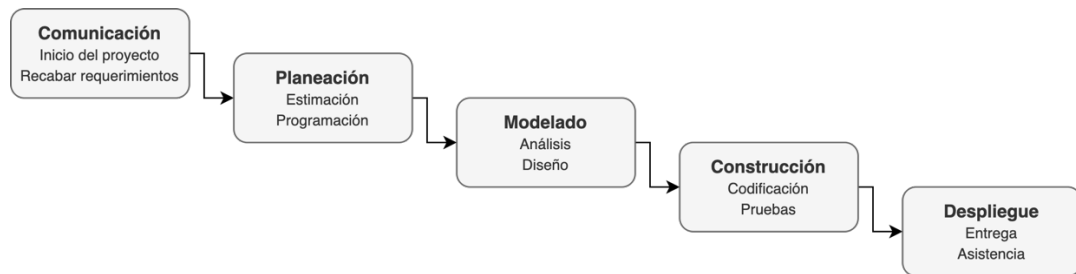
La ingeniería de software se puede definir como la aplicación de un enfoque sistemático, disciplinado, adaptable, ágil y cuantificable al desarrollo, operación y mantenimiento de software (Pressman & Troya, 2010). Se trata de un concepto que fue mencionado por primera vez en la década de 1960, en una conferencia donde se trataron temas referentes a la crisis que el desarrollo de software enfrentaba en su momento (software costoso de desarrollar, mantener y de baja calidad).

A partir de su concepción, se diseñaron y probaron distintos modelos para el desarrollo de software. Todos estos, tenían un objetivo en común, la generación eficiente de software fácil de mantener y de calidad. Con este en mente, la ingeniería

de software se sustentó en cuatro compromisos esenciales: utilizar de herramientas genéricas, aplicar estándares, definir procesos y asegurar la calidad del producto final (Jabangwe et al., 2018; Pressman & Troya, 2010). A partir de estos compromisos, fueron desarrollados modelos de desarrollo de software que permitieran cumplir con los cuatro sustentos de esta disciplina. El primero de estos modelos, fue el modelo en cascada, que sirvió como una representación de cómo es que se realizaba el proceso de desarrollo de software en ese momento (Figura 2.1).

Figura 2.1.

Modelo en cascada original



Nota. Elaboración propia con base en Pressman & Troya (2010).

Del modelo en cascada, se derivaron múltiples modelos de desarrollo de software que poseían características propias según el contexto y tipo de proyectos de desarrollo en el que se esperaba que fueran utilizados (Heeager & Nielsen, 2018; Pressman & Troya, 2010). Así fueron creados modelos muy populares como: modelo en V, modelo de desarrollo incremental/espiral y el modelo orientado a prototipos. Estos modelos, sin embargo, resultarían insuficientes para cumplir los compromisos que dan sustento a la ingeniería de software.

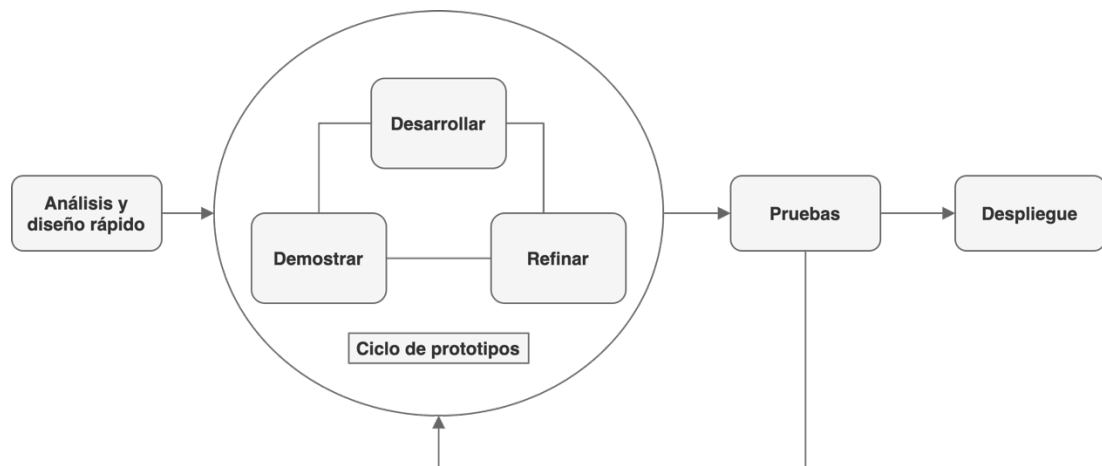
Fue hasta la década de 1970, cuando se crearon y se introdujeron métricas para la estimación de costos y de tiempos en los proyectos de desarrollo, que la ingeniería de software se consolidó como disciplina (Arvanitou et al., 2016). Con la inclusión de las métricas, la definición del ciclo de vida del software y la creación de lenguajes como UML, los modelos evolucionaron, convirtiéndose en metodologías

que buscaban que cualquiera (sin importar su experiencia) pudiese utilizarlas y cumplir con los objetivos de la ingeniería de software (Pressman & Troya, 2010).

Fue de esa manera como surgieron algunas metodologías de desarrollo de software tradicionales, como lo es RAD (*Rapid Application Development*). Estas metodologías con más complejas que los modelos originales, esto, al ser conjuntos de modelos que no presentan una serie establecida de pasos, sino que presentan una serie de actividades y roles que deben cumplirse para la creación de software de calidad (Figura 2.2) (Angelov & de Beer, 2017).

Figura 2.2.

Metodología RAD



Nota. Elaboración propia con base en Williams (2010).

No obstante, pese a la consolidación de modelos en metodologías de desarrollo, aún existían problemas relacionados con la programación. Aunque el proceso se encontraba bien definido y permitía reducir costos y tiempos, aún existían problemas de fondo en la programación del software. Debido a esto, surgieron la arquitectura de software y los patrones de diseño de software (PDS), mismos que darían solución a estos problemas (Bass et al., 2013; Gamma et al., 1995; Richards, 2015). Entonces, la Ingeniería de Software tiene un papel importante cuando se habla del desarrollo de plataformas de tecnología educativa, al ser la disciplina que, en principio, rige el proceso de desarrollo de software mediante el que se genera una plataforma educativa.

2.2. Diseño Instruccional

El diseño instruccional es una disciplina que cambia y evoluciona constantemente según el contexto en el que se desarrolla y las circunstancias en las que se da el proceso de enseñanza-aprendizaje. Así pues, han existido múltiples modelos de diseño instruccional que representan la concepción que se tenía de esta disciplina en un determinado momento (Belloch, 2017). Estos modelos se encuentran directamente relacionados con la teoría de aprendizaje predominante en su momento. En la Tabla 2.1, se presentan las características del diseño instruccional según la época.

Tabla 2.1.

Contexto histórico del diseño instruccional

Década	Características	Modelos	Teoría de aprendizaje
1960	Lineal, sistemático y prescriptivo. Se centra en los conocimientos y destrezas académicas observables y medibles.	Dick y Carey	Conductismo
1970	Organizado en sistemas abiertos. Busca una mayor participación de los estudiantes, al quitar ciertas responsabilidades al docente.	Gagné y Briggs	Sistemas
1980	Se preocupa por la comprensión de los procesos de aprendizaje, con énfasis en los procesos cognitivos como el pensamiento, la resolución de problemas, el lenguaje y la formación de conceptos.	-	Cognitivismo
1990	Convierte en actor principal al aprendiz (estudiante), por lo que las acciones formativas deben centrarse en el proceso de aprendizaje y en la creatividad del estudiante.	ADDIE, ASSURE, Jonassen	Constructivismo y conectivismo

Nota. Elaboración propia con base en Belloch (2017).

Entonces, el diseño instruccional se puede definir como una ciencia encargada de la planeación, diseño, creación e implementación de una instrucción de una forma eficiente y efectiva. Lo anterior, implica que el diseño instruccional

contempla todos los procesos que permiten optimizar el aprendizaje y el desempeño de los estudiantes (Sharif & Cho, 2015).

Tal como sucedía en el pasado, en la actualidad el diseño instruccional se encuentra en un constante cambio. En esta ocasión, este cambio se da en respuesta a la creciente inclusión de la tecnología en los procesos de enseñanza-aprendizaje tradicionales. Sin importar de qué modalidad se hable, sea *e-learning*, *b-learning* o incluso *u-learning*, el diseño instruccional permite plantear un proceso sistémico con actividades interrelacionadas y multidisciplinarias que faciliten la construcción del conocimiento.

Es con la premisa anterior que surge el concepto de diseño tecnoinstruccional, tecnopedagógico o diseño instruccional digital (Belloch, 2017; Campos et al., 2019; Gros et al., 2016). En el cual, aparte de la dimensión pedagógica, existe una dimensión tecnológica y didáctica, misma que incrementa drásticamente las habilidades, competencias y conocimientos requeridos por aquellos que deseen utilizar el diseño instruccional en su labor docente (Ritzhaupt & Kumar, 2015). Lo anterior, es motivo de múltiples investigaciones, ya que implica que el uso de tecnología en el aprendizaje tiene una mayor profundidad que lo que se creía y, por lo mismo, trae consigo retos que antes no habían sido considerados (Nori et al., 2014). La inclusión, accesibilidad, la educación integral, las competencias digitales tanto de docentes como de estudiantes, el potencial del trabajo remoto y la incertidumbre son solo algunos de los retos que se presentan en el proceso de enseñanza-aprendizaje.

2.3. Proceso de enseñanza-aprendizaje

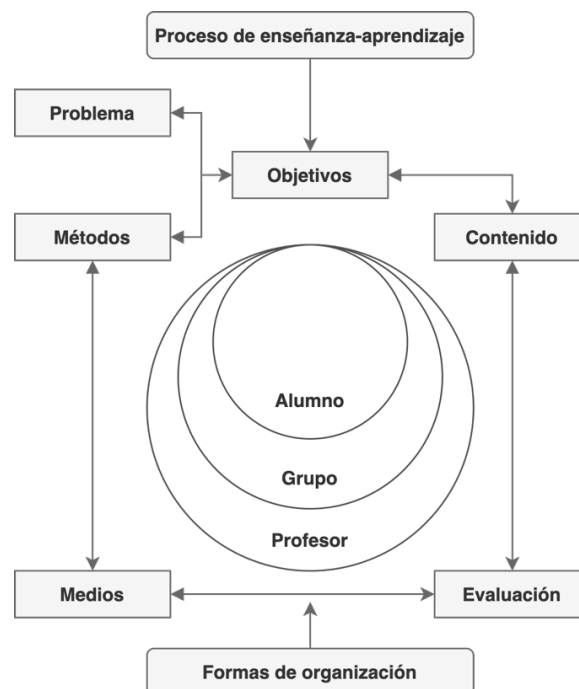
La enseñanza y el aprendizaje son dos conceptos cuyo estudio es tan antiguo como el del ser mismo. Disciplinas como la filosofía y psicología han estudiado constantemente el proceso a través del cual el ser humano aprende y genera conocimiento (Holt & Kysilka, 2005).

El aprendizaje debe ser estudiado a través de la enseñanza y viceversa, resulta prácticamente imposible hablar de un concepto sin tratar al otro. El proceso de enseñanza-aprendizaje puede definirse, en un principio, como un sistema de comunicación intencional entre un aprendiz y un instructor, que se produce en un marco institucional y en el que se generan estrategias para promover el aprendizaje (Meneses, 2007). Esta definición, sin embargo, obedece a un contexto socioconstructivista en el cual no existen de forma inherente conceptos como el *e-learning* o el *u-learning*. Tal como sucede con el diseño instruccional, el proceso de enseñanza-aprendizaje ha evolucionado y cambiado debido a la presencia cada vez más ubicua de la tecnología (Rodríguez, 2014).

Se trata de un proceso que cuenta con múltiples componentes que actúan en conjunto y en diferentes niveles. Estos componentes han variado con el paso del tiempo y según el contexto de las investigaciones que se encargaron de estudiarlo en su momento (Figura 2.3).

Figura 2.3.

Componentes del proceso de enseñanza-aprendizaje



Nota. Elaboración propia con base en García (2020).

Entonces, debido a la introducción de tecnología en las aulas, a la generación de espacios para el aprendizaje informales y al surgimiento de conceptos como la tecnología educativa, la definición de este proceso cambió. Ya no se trata únicamente sobre cómo se aprende, sino que ahora se contemplan aspectos como el quién y cuándo se aprende (Rodríguez, 2014). Lo anterior, ha conducido a una serie de cambios donde, al igual que en el diseño instruccional, ya no existe únicamente una dimensión pedagógica y una dimensión de conocimiento, sino que ahora existe y es necesario considerar una dimensión tecnológica, como la propuesta en modelos como el TPACK (*Technological Pedagogical Content Knowledge*) (Koehler & Mishra, 2005).

3. Marco teórico

3.1. Tecnología educativa

Actualmente, las Tecnologías de la Información y Comunicación (TIC) se encuentran presente en prácticamente todos los ámbitos referentes a la educación. Nunca un docente había tenido tantas herramientas disponibles a su alcance para la enseñanza, la administración educativa y la investigación (Cabero, 2015). Esta presencia tan predominante de las TIC en la educación ha conllevado la creación del término tecnología educativa, mismo que cuenta con definiciones muy diversas.

La tecnología educativa constituye la disciplina encargada del estudio de los medios, materiales, portales web y plataformas tecnológicas al servicio de los procesos de aprendizaje, en cuyo campo se encuentran los recursos aplicados con fines formativos e instruccionales, diseñados originalmente como respuesta a las necesidades e inquietudes de los usuarios (Serrano et al., 2016).

No obstante, también es posible definirla como el uso pedagógico del conjunto de herramientas, procedimientos y técnicas para mejorar los procesos de enseñanza y aprendizaje. La tecnología educativa crea, utiliza y administra procesos tecnológicos y recursos educativos para ayudar a mejorar el rendimiento académico de los usuarios (Camarda & Minzi, 2012).

Entonces, es posible definir el concepto de tecnología educativa como el uso combinado de hardware, software, teoría y práctica educativa para facilitar tanto el aprendizaje como la enseñanza. Esto implica, que esta se encarga de diseñar, crear, utilizar y administrar los procesos y recursos tecnológicos para ayudar a mejorar el proceso de enseñanza-aprendizaje (Robinson et al., 2013). De lo anterior, es posible mencionar que la tecnología educativa también se trata del espacio intelectual pedagógico, mismo que tiene como objeto de estudio los medios y las tecnologías de la información, en cuanto formas de representación, difusión y acceso al conocimiento y a la cultura en los distintos contextos educativos (Area, 2009).

La investigación sobre las aplicaciones de las TIC en la enseñanza comenzó en Estados Unidos, a partir de las cuales surgieron diversos enfoques que son los responsables de la diversidad de definiciones existentes al respecto. En la Tabla 3.1 se presentan los enfoques teóricos bajo los cuales se estudia la tecnología educativa.

Tabla 3.1.

Enfoques de investigación sobre tecnología educativa

Enfoque	¿Qué investiga?
Conductista	- Efectos del tipo de medios sobre el rendimiento.
Cognitivista	- Efectos de los atributos de los medios sobre el procesamiento cognitivo y/o el rendimiento.
	- Efectos del método instruccional sobre el procesamiento cognitivo y el rendimiento.
	- Los efectos de la interacción entre las aptitudes del estudiante y el método instruccional sobre el proceso cognitivo y el rendimiento.
Conectivista	- El aprendizaje en la diversidad de opiniones. Las redes de conocimiento en el aprendizaje.
Constructivista	- La interacción dinámica entre los distintos actores (docentes/estudiantes).
	- La construcción del conocimiento a partir de la interacción.

Nota. Elaboración propia con base en Ferrer & Madriz (2009) y Oropeza et al. (2016).

Si bien son solo algunos de los enfoques existentes, queda claro que están centrados en aplicaciones concretas del campo educativo y no necesariamente en uno tecnológico (Ferrer & Madriz, 2009). Lo anterior representa un campo de dominio completamente diferente sobre el cuál se puede desarrollar el área de tecnología educativa, donde se contempla no solo por qué y para qué del proceso de enseñanza-aprendizaje mediado por TIC, sino el cómo se puede implementar dicho proceso desde un punto de vista técnico (Gros, 2012).

Así pues, de la tecnología educativa emergieron múltiples conceptos que consideran no solo el proceso de enseñanza-aprendizaje, sino su implementación

y las implicaciones que la inclusión de la tecnología conlleva en este proceso (ver Tabla 3.2).

Tabla 3.2.

Modalidades afines a la tecnología educativa

Conceptos	Descripción	Caracterización
Educación a distancia	Sistema tecnológico de comunicación bidireccional. Puede ser masivo, con base en la acción sistemática y conjunta de recursos didácticos mediados por la tecnología.	Modalidad educativa no presencial. Solo recientemente comenzó a apoyarse de medios electrónicos.
Educación virtual	Proceso educativo no continuo e intangible, permite técnicamente el fraccionamiento de sus componentes en distintos ambientes virtuales.	Pese a su similitud con la educación a distancia, aquí son más constantes y frecuentes las interacciones profesor-alumno y alumno-alumno.
<i>e-learning</i>	Utilización de tecnologías de internet para el aprendizaje síncrono y asíncrono mediante plataformas de enseñanza-aprendizaje.	Modalidad educativa usualmente mediada o apoyada por herramientas tecnológicas en actividades en las que se requiere de la intervención de docentes y estudiantes.
<i>b-learning</i>	Aprendizaje mixto, donde lo virtual se complementa con lo presencial (mixto) y se emplean las características de ambos procesos de enseñanza.	En esta modalidad existe una clara separación de los momentos y espacios virtuales y presenciales, a diferencia del e-learning donde estos pueden ocurrir de forma simultánea.
<i>u-learning</i>	Escenario formativo, el cual es un conjunto de actividades formativas apoyadas en las tecnologías que permite el aprendizaje en cualquier momento, lugar y en cualquier dispositivo de información actual.	Contempla la educación mediante dispositivos móviles y el aprendizaje autónomo.

Nota. Elaboración propia con base en Jiménez-Saavedra (2014).

Entonces, de los conceptos afines, es posible identificar que el principal diferenciador entre estos es el grado de inclusión, integración y protagonismo de las tecnologías de la información. Sin embargo, no deja de ser difusa la línea entre estos conceptos y, por lo mismo, tampoco resulta clara la división de

responsabilidades sobre quién (o quiénes recae) el desarrollo cursos que hacen uso de medios digitales y tecnológicos (Ortega Barba, 2014). Esto último, es la razón por la que disciplinas como el diseño instruccional, buscan facilitar este proceso de integración tecnológica (Belloch, 2017).

3.2. Diseño instruccional

El desarrollo de cualquier curso de instrucción debe seguir un proceso, ya sea consciente o rutinario, para alcanzar los objetivos educativos planeados (Belloch, 2017). El proceso habitual para lograr lo anterior se llama diseño instruccional.

El diseño instruccional ha sido definido de múltiples formas, cada una con un énfasis diferente, de acuerdo con diferentes posturas conceptuales. Una de estas posturas establece que el diseño instruccional es el proceso encargado de la planeación, preparación y el diseño de los recursos y ambientes necesarios para que se desarrolle el aprendizaje (Bruner, mencionado en Belloch, 2017).

Por otra parte, es posible definirlo como la disciplina encargada de crear un ambiente instruccional, así como al conjunto de los materiales que ayudarán al alumno en su proceso de adquisición del conocimiento (Broderick, mencionado en Campos et al., 2019). Mientras que, en complemento a lo anterior, se puede decir que el diseño instruccional se trata del proceso de desarrollo de una clase, mediante la aplicación de métodos, técnicas y estrategias didácticas (Belloch, 2017). En la Tabla 3.3, se presentan algunos de los modelos de diseño instruccional más conocidos.

Tabla 3.3.

Modelos de diseño instruccional

Modelo	Descripción	Fases
ADDIE	Se trata del modelo más dinámico. Tiene como propósito la formación de individuos para el desarrollo de un proyecto en específico.	<ol style="list-style-type: none"> 1. Análisis 2. Diseño 3. Desarrollo 4. Implementación 5. Evaluación

Modelo	Descripción	Fases
ASSURE	Con raíces teóricas en el constructivismo, este modelo parte de las características del estudiante y de su estilo de aprendizaje para fomentar la participación y el compromiso de los estudiantes.	<ol style="list-style-type: none"> 1. Analizar a los estudiantes 2. Fijar objetivos 3. Seleccionar los métodos de formación 4. Utilizar los medios y materiales 5. Exigir la participación de los alumnos 6. Evaluar y revisar
Jonassen	Modelo para ambientes de aprendizaje constructivistas que enfatiza el papel del estudiante en su proceso de adquisición del conocimiento.	<ol style="list-style-type: none"> 1. Preguntas, casos y problemas 2. Casos relacionados 3. Recursos de información 4. Herramientas cognitivas 5. Conversación/Herramientas de colaboración 6. Social/Apoyo del contexto
Dick, Carey y Carey	Fundamentado en la idea de una relación predecible y fiable entre los materiales didácticos y el aprendizaje. Recae en el diseñador la tarea de identificar las competencias que alumno debe dominar y, a su vez, mediante qué actividades estimulará su adquisición.	<ol style="list-style-type: none"> 1. Identificar meta instruccional 2. Análisis de la instrucción 3. Análisis de los estudiantes y el contexto 4. Redacción de los objetivos 5. Desarrollo de instrumentos de evaluación 6. Elaboración de la estrategia instruccional 7. Desarrollo y selección de materiales de instrucción 8. Diseño y desarrollo de la evaluación formativa 9. Diseño y desarrollo de la evaluación sumativa 10. Revisión de la instrucción
Gagné y Briggs	Este modelo fundamentado en los procesos cognitivos propone una aproximación con base en el enfoque de sistemas.	<ol style="list-style-type: none"> 1. Nivel de sistema 2. Nivel de curso 3. Nivel de la lección 4. Nivel de sistema final

Nota. Elaboración propia con base en López (2018).

Así pues, al tratarse de un proceso, el diseño instruccional ha sido modelado por múltiples investigaciones, con el objetivo de facilitar la reutilización de escenarios y fomentar la colaboración en la creación de entornos de aprendizaje en

distintos contextos (Gros et al., 2016). Lo anterior, implicaría la existencia de modelos específicos para el desarrollo del proceso de enseñanza-aprendizaje con el uso de tecnología educativa o en ambientes virtuales de aprendizaje, sin embargo, este no ha sido el caso (Domínguez et al., 2020; Luna-Gijón & Porrás-Hernández, 2014).

Si bien, se han adaptado modelos de diseño instruccional para su uso con la tecnología educativa, es una realidad que ninguno ha sido desarrollado de forma específica con este fin. Lo anterior, ha resultado en la aplicación inadecuada de múltiples modelos de diseño instruccional en entornos virtuales y en el *e-learning*. Esto, en virtud de que, pese a la existencia de múltiples modelos, la mayoría tienen como base la misma serie de principios, lo que genera confusión y una mala comprensión de los mismos (Adnan & Ritzhaupt, 2018). Debido a esto, el diseño instruccional se ha convertido en una disciplina que cambia de forma rápida y constante (Parmaxi et al., 2016).

Por lo tanto, el surgimiento de conceptos derivados de la tecnología educativa como el *e-learning* y el *u-learning*, al ser términos que unen la tecnología y los medios digitales con la educación, han generado la necesidad de diseñadores instruccionales que no solo cuenten con bases sólidas de teorías del aprendizaje sino que también cuenten con competencias y habilidades tanto digitales como técnicas (como el proceso de desarrollo de software) (Adnan & Ritzhaupt, 2018; Ritzhaupt & Kumar, 2015).

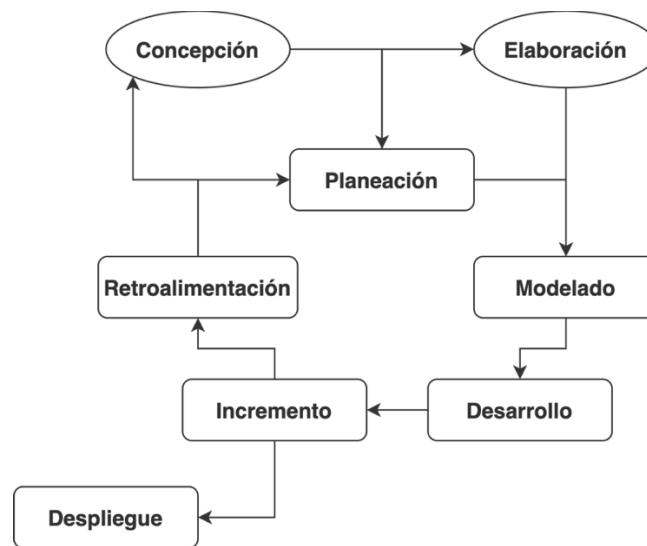
3.3. Proceso de desarrollo de software

El desarrollo de software es una labor que, en sí misma, presenta varios y diversos retos, pese a toda la atención e investigación que ha sido destinada a la misma (Angelov & de Beer, 2017). La realidad que enfrentan los equipos de desarrollo se encuentra llena de retos que implica el trabajo distribuido, diferentes niveles de conocimientos, factores organizacionales y tiempos cortos para proyectos de gran tamaño que derivan en otra serie de problemas como una pobre documentación (Papadopoulos, 2015).

Con la finalidad de resolver las problemáticas planteadas, surgió la ingeniería de software, cuyos principios y conceptos se encuentran plasmados en diferentes metodologías de desarrollo de software (Pressman & Troya, 2010). Estas metodologías surgen a partir de modelos básicos de desarrollo de software y sirven como las estructuras para las actividades y tareas que se requieren a fin de construir software de calidad.

Los modelos básicos de desarrollo de software son: modelo de la cascada, modelo de proceso incremental, modelo de proceso evolutivo y modelo concurrente. A partir de estos, surgieron las metodologías tradicionales para el desarrollo de software, de las cuales la más representativa es el Proceso Racional Unificado (RUP, por las siglas en inglés de *Rational Unified Process*) (Pressman & Troya, 2010). En la Figura 3.1 se presentan las actividades que desarrolla la metodología RUP.

Figura 3.1.
Metodología RUP



Nota. Elaboración propia con base en Pressman & Troya (2010).

A partir de RUP, emergieron las metodologías de desarrollo de software ágiles como una evolución natural del proceso de desarrollo de software. Estas, fueron creadas durante la segunda mitad de la década de los noventa y

formalizadas en el año 2001 (Özcan-Top & Demirors, 2019). Las metodologías para el desarrollo de software ágil tienen como propósito presentar una alternativa a los procesos y técnicas tradicionales (como RUP), con énfasis en la rapidez del proceso. Las metodologías reconocidas por adoptar este principio fundamental son: *Scrum*, *Crystal*, DSDM y Programación Extrema (XP, del inglés *Extreme Programming*) (Özcan-Top & Demirors, 2019; Williams, 2010).

Todas estas metodologías contrastan con la metodología RUP, ya que optan por darle prioridad al desarrollo en lugar de a la documentación y diseño de un proyecto. Lo anterior, permite reducir el número de los involucrados, presentar avances significativos en cortos períodos de tiempo y disminuir los tiempos de adaptación para los integrantes del equipo de desarrollo al no requerir de una capacitación tan extensa (Delgado, 2008). En la Tabla 3.4 se presentan las características propias de las metodologías de desarrollo ágil.

Tabla 3.4.

Descripción de metodologías ágiles

Metodología	Descripción
Crystal	Centrada en la comunicación dentro de equipos pequeños, se caracteriza por tener entregas frecuentes, mejoras reflexivas, seguridad personal, enfoque y un nivel de entrada bajo.
DSDM	Divide el proyecto en tres fases: pre- proyecto, ciclo de vida del proyecto y post- proyecto.
Scrum	Centrado en la administración de proyectos donde existe la dificultad de planear a futuro. El software es desarrollado por equipos auto gestionados. El equipo comienza con la planeación y termina con la revisión.
XP	En esta metodología prevalecen las mejores prácticas de programación. El software se documenta a la par que se realiza, por lo que existen tiempos de desarrollo muy cortos.

Nota. Elaboración propia con base en Dybå & Dingsøyr (2008).

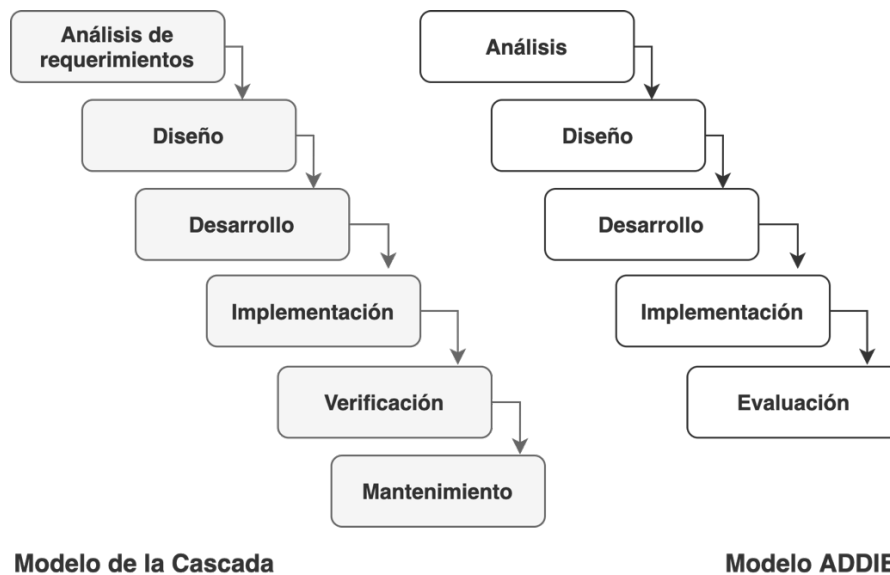
Las metodologías ágiles facilitan y vuelven más accesible el proceso de desarrollo, sin embargo, tampoco se encuentran libres de problemas y retos (Özcan-

Top & Demirors, 2019), mismos que cada equipo de desarrollo (según el proyecto que se desarrolle) afronta de la manera que mejor se ajuste a sus necesidades. Ahora bien, el desarrollo de plataformas de tecnología educativa, al igual que todos, cuenta con necesidades y requerimientos específicos que exigen un enfoque multidisciplinario (Chimalakonda & Nori, 2018). En particular, al analizar metodologías tradicionales o ágiles, resulta complicado pensar en que sean empleadas por diseñadores instruccionales o por profesores, ya que no cuentan con los conocimientos técnicos necesarios (Laurillard et al., 2018).

Así pues, existen algunas iniciativas y propuestas que buscan homologar los procesos establecidos en los modelos de diseño instruccional y aquellos definidos por la ingeniería de software para el desarrollo de productos (Figura 3.2) (Adnan & Ritzhaupt, 2018). No obstante, lo anterior ha resultado ser una tarea compleja dadas las diferencias semánticas entre ambas áreas del conocimiento.

Figura 3.2.

Comparativa de modelos de diseño instruccional y de desarrollo de software



Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018).

Entonces, ¿qué solución podría existir para la creación de un modelo de desarrollo de software orientado a las plataformas de tecnología educativa?, en

respuesta a esto, se ha planteado el uso de patrones arquitectónicos y de diseño de software, ya que aunque son elementos propios de la ingeniería de software, utilizan un lenguaje no tan técnico y que podría empatar con lo utilizado en el diseño instruccional (Chimalakonda & Nori, 2018; Nori et al., 2014).

3.4. Arquitectura de software

Los sistemas de software complejos pueden ser descritos como una composición de elementos que interactúan entre sí. Dichas interacciones ocurren por la necesidad de emplear características presentes en otros componentes y por el intercambio de datos. Entonces, cambiar un solo componente puede tener un impacto significativo en todo el sistema. La arquitectura de software, permite representar de forma concreta y concisa los componentes y sus interacciones para contar con una visualización gráfica de alto nivel de un sistema de software (Bjuhr et al., 2017).

La arquitectura de software se puede definir como el conjunto de estructuras necesarias para razonar sobre un sistema. Comprende elementos de software, las relaciones entre éstos y las propiedades de los elementos y sus relaciones (Bass et al., 2013). También es posible describirla como la identificación y entendimiento de cada componente de la solución (software) y la descripción de como interactúan dichos componentes para satisfacer una serie de requerimientos. Es de esta manera, que la arquitectura de software se consolida como un aspecto de suma importancia al momento de realizar una toma de decisiones en cualquier proceso de desarrollo de software (van Vliet & Tang, 2016).

Propuestos por Buschmann et al. (2001), los patrones arquitectónicos surgen con la finalidad de facilitar la identificación de arquitecturas y de proveer a los desarrolladores con conceptos familiares que les permitan diseñar una arquitectura correcta. Al igual que con los PDS, han sido ampliamente adoptados por la ingeniería de software, a la vez que su implementación y análisis aún es motivo de investigación. En la Tabla 3.5, se presentan la clasificación de patrones arquitectónicos y su descripción.

Tabla 3.5.

Patrones arquitectónicos de software

Clasificación	Descripción
<i>From Mud to Structure</i>	Estos patrones permiten evitar grandes cantidades de componentes y objetos. En particular, apoyan la descomposición de la labor general de un sistema en subtareas. Los patrones de esta categoría son: <i>Layers, Pipes and Filters</i> y <i>Blackboard</i> .
Sistemas Distribuidos	Esta categoría incluye únicamente un patrón, <i>Broker</i> , que provee de una infraestructura estandarizada para aplicaciones distribuidas.
Sistemas Interactivos	Esta categoría está compuesta de dos patrones, <i>Model-View-Controller</i> y <i>Presentation-Abstraction-Control</i> . Ambos dedicados a dar soporte a la estructura de sistemas que se caracterizan por la interacción humano-computadora.
Sistemas Adaptables	Conformada por los patrones <i>Reflection</i> y <i>Microkernel</i> , apoyan fuertemente la extensión de aplicaciones y su adaptación a tecnologías evolutivas y requerimientos funcionales cambiantes.

Nota. Elaboración propia con base en Buschmann et al. (2001).

Contrario a lo que parecería, los patrones arquitectónicos de software fueron pensados para que sean entendidos por todos los interesados en un producto de desarrollo de software (Bass et al., 2013; Richards, 2015). Lo anterior, implica la posibilidad de utilizarlos en un ámbito distinto al de la ingeniería de software, ya que su comprensión no depende de conceptos tan técnicos, únicamente requieren de una definición clara de las necesidades de los usuarios finales del producto. Entonces, su uso por parte de diseñadores instruccionales es factible, sin embargo, estos no quedan exentos de contar las competencias digitales suficientes para comprender las implicaciones que los requerimientos tienen sobre el desarrollo de una plataforma, tanto en lo no funcional como en lo funcional (donde se encuentran los patrones de diseño de software).

3.5. Patrones de diseño de software

Los PDS son soluciones probadas y efectivas a problemas recurrentes de diseño durante el proceso de desarrollo de software (Riaz, 2018). Impiden que, tanto desarrolladores como arquitectos de software, tengan que reinventar soluciones

óptimas, ahorrando tiempo valioso para el desarrollo. Fueron propuestos por Gamma et al. (1995), quienes los definen como una solución presente en el núcleo de un problema recurrente que, por cuya naturaleza, es posible reutilizar una y otra vez sin aplicarla necesariamente de la misma forma dos veces. Básicamente, proponen soluciones que obedecen a determinados contextos, mas no a dominios específicos.

Desde su concepción, han sido ampliamente adoptados y aplicados como buenas prácticas en el proceso de desarrollo de software. Son considerados como una parte medular de la ingeniería de software (Fourcade et al., 2017). Así como sucedió con las heurísticas de usabilidad propuestas por Jakob Nielsen, los patrones de diseño se han convertido en referentes vigentes utilizados en la industria, ya que son una herramienta que trasciende generaciones al encapsular la experiencia y conocimiento de los desarrolladores en un lenguaje común y aplicable en diversas situaciones (Khomh & Gueheneuc, 2018).

Cada patrón se encuentra compuesto por cuatro elementos esenciales: Nombre, Problema, Solución y Consecuencia. En la Tabla 3.6, se presenta una descripción más detallada de la composición de los PDS.

Tabla 3.6.

Composición de los patrones de diseño

Elemento	Descripción
Nombre	Es la descripción del problema de diseño, su solución y la consecuencia en una palabra o no. Es una manera de ampliar el vocabulario de diseño a un nivel más alto de abstracción.
Problema	Describe cuándo aplicar el patrón. Explica la problemática y su contexto, puede describir problemas de diseño específicos tales como la manera de representar un algoritmo o describir estructuras con diseños poco flexibles. Puede incluir una lista de condiciones que se deben de cumplir para que el patrón pueda ser aplicado.

Elemento	Descripción
Solución	Describe los elementos que componen: el diseño, sus relaciones, responsabilidades y dependencias. No describe un diseño en concreto o una implementación, debido a que un patrón es una plantilla aplicable en múltiples contextos.
Consecuencia	Presenta los resultados y concesiones que conllevan la aplicación de un patrón. Incluye el impacto en la flexibilidad, extensibilidad, portabilidad y atributos de calidad de un sistema.

Nota. Elaboración propia con base en Gamma et al. (1995).

A partir de estos elementos se conformaron 23 patrones de diseño, los cuales cuentan con diferentes grados de complejidad y se encuentran clasificados, según el contexto al que pertenecen, en: creacionales, estructurales y conductuales (Hussain et al., 2017). Los patrones de diseño creacionales están enfocados en el proceso de creación de objetos, tienen como propósito desacoplarse de la implementación del sistema (Tirado et al., 2019). Existen cinco patrones creacionales, los cuales se encuentran descritos en la Tabla 3.7.

Tabla 3.7.

Patrones de diseño creacionales

Patrón	Descripción
<i>Abstract Factory</i>	Provee de una interfaz donde se crean familias de objetos relacionados o dependientes sin la necesidad de especificar su clase concreta.
<i>Builder</i>	Separa la construcción de un objeto complejo de su representación. Es decir, separa la estructura de la funcionalidad.
<i>Factory Method</i>	Define una interfaz para crear un objeto, permite a las clases diferir la instanciación a las subclases.
<i>Prototype</i>	Especifica tipos de objetos a crear utilizando instancias prototípicas, creando nuevos objetos a partir de dicho prototipo.
<i>Singleton</i>	Asegura que una clase tenga una única instancia y provee un punto de acceso global a esta.

Nota. Elaboración propia con base en Riaz (2018).

Por su parte, un patrón estructural es aquel que hace uso de la herencia para componer interfaces o implementación, describe la forma en que se componen objetos para

obtener nuevas funcionalidades a la vez que se le brinda flexibilidad al sistema (Peñalvo, mencionado en Tirado et al., 2019). Los patrones de diseño estructurales se encuentran descritos en la Tabla 3.8.

Tabla 3.8.

Patrones de diseño estructurales

Patrón	Descripción
<i>Adapter</i>	Convierte la interfaz de una clase en una interfaz esperada por otro componente de software, permite el trabajado colaborativo de las clases pese a su incompatibilidad.
<i>Bridge</i>	Desacopla una abstracción de su implementación para que ambas puedan operar de forma independiente.
<i>Composite</i>	Compone objetos en estructuras de árbol para representar jerarquías todo-parte.
<i>Decorator</i>	Añade responsabilidades (funcionalidades) adicionales a un objeto de manera dinámica.
<i>Facade</i>	Provee de una interfaz unificada a un conjunto de interfaces en un sistema. Define la interfaz de más alto nivel que facilita el uso del sistema.
<i>Flyweight</i>	Usa la división para soportar grandes números de objetos de manera eficiente.
<i>Proxy</i>	Provee de un sustituto para otro objeto para controlar su acceso.

Nota. Elaboración propia con base en Riaz (2018).

Por último, los patrones conductuales son aquellos que se caracterizan por la forma en que los objetos o clases interactúan y distribuyen las responsabilidades entre sí (Hussain et al., 2017). Dichos patrones se encuentran descritos en la Tabla 3.9.

Tabla 3.9.

Patrones de diseño conductuales

Patrón	Descripción
<i>Chain of Responsibility</i>	Evita el acoplamiento del emisor de una petición al receptor, permitiendo que más de un objeto pueda manejar dicha petición.

Patrón	Descripción
<i>Command</i>	Encapsula la petición en un objeto, permitiendo parametrizar los clientes con diferentes solicitudes.
<i>Interpreter</i>	Dado un lenguaje de programación, describe su gramática en conjunto con un intérprete para interpretar sentencias.
<i>Iterator</i>	Provee de un acceso a los elementos de un objeto agregado de manera secuencial sin exponer su representación.
<i>Mediator</i>	Define un objeto que encapsula como un conjunto de interactúa. Promueve un bajo acoplamiento al permitir la comunicación explícita entre objetos.
<i>Memento</i>	Permite externalizar el estado interno de un objeto sin romper la encapsulación de este.
<i>Observer</i>	Define una dependencia de uno-a-muchos entre objetos para que, cuando uno cambie de estado, todos sus dependientes actúen en consecuencia.
<i>State</i>	Permite a un objeto alterar su comportamiento cuando su estado interno cambia.
<i>Strategy</i>	Define una familia de algoritmos que son encapsuladas e intercambiables.
<i>Template Method</i>	Define el esqueleto de un algoritmo en una operación, difiriendo etapas a subclases propias.
<i>Visitor</i>	Representan una operación ejecutada en los elementos estructurales de un objeto.

Nota. Elaboración propia con base en Riaz (2018).

Así pues, los PDS sirven como una base de cualquier proyecto de desarrollo de software, ya que proveen de una estructura fiable a los mismos (Chitra, 2018). Este mismo acercamiento ha sido adoptado por otras disciplinas, como en la educación, donde se han identificado patrones en el aprendizaje, mismos que permiten una mejor comprensión de la integración de las TIC en los procesos de enseñanza-aprendizaje (Aguña, 2012). De igual manera, esta misma noción fue adoptada por el diseño instruccional, una disciplina que podría servir de puente entre la brecha presente entre la teoría y la práctica de la tecnología educativa (Lowyck, 2014; Nori et al., 2014).

3.6. Diseño de software educativo

Diseñar software se trata de una tarea colaborativa y multidisciplinaria. En el contexto de la plataformas de tecnología educativa, se trata de una actividad que requiere la participación de diferentes áreas de conocimiento como la ingeniería de software y el diseño instruccional (Martinez-Maldonado et al., 2017). No obstante, pese a la existencia de múltiples técnicas y herramientas para el diseño tanto de software como instruccional, la realidad es que no existe una definición clara sobre lo que implica en realidad el diseño de software educativo.

Por una parte, se podría pensar que esta actividad recaee de forma exclusiva en un apartado técnico, como es el caso del diseño de interfaces de usuario o el diseño arquitectónico de una plataforma. Sin embargo, esto no es así, ya que se trata de una actividad que también debe contemplar el diseño de actividades, cursos, materiales y objetos de aprendizaje (Exter, 2018; McKenney et al., 2015). Esto, aunque suene bien en la teoría, en la práctica resulta complejo, ya que la gestión de todas estas actividades no solo requiere de una planeación concreta y extensa, sino que también se requiere de personas con una formación en áreas diametralmente opuestas como lo son la pedagogía, diseño instruccional e ingeniería de software (Nori et al., 2014; Ritzhaupt & Kumar, 2015).

Pese a lo anterior, el estudio y la búsqueda de una solución a esta problemática han iniciado la definición (aunque sea de forma incipiente) las responsabilidades que cada área del conocimiento tiene en el proceso de diseño y desarrollo de plataformas de tecnología educativa (San Martín & García, 2016). Así pues, al hablar de conceptos como *e-learning*, *u-learning* y *b-learning*, ya no implica solo el estudio de los procesos de enseñanza-aprendizaje en diferentes medios, ahora se integran características y aspectos que antes pertenecían de forma exclusiva a la ingeniería de software (Angelov & de Beer, 2017; Tai & Ting, 2020). Esto, se puede apreciar en la irrupción tecnológica en el diseño instruccional y en conceptos como la tecno-pedagogía (Lorenzo-Lledó, 2018).

En este sentido, en la actualidad, el diseño de software educativo necesita contemplar dos aspectos: pedagogía y tecnología. En la Tabla 3.10 se presentan las características de estos aspectos.

Tabla 3.10.

Aspectos necesarios en el diseño de software educativo

Característica	Aspecto pedagógico	Aspecto tecnológico
Objeto de estudio	Diseño instruccional, guiones tecno-pedagógicos, objetos de aprendizaje, etc.	Arquitectura del sistema, características suplementarias (usabilidad, modularidad, portabilidad, etc.), diseño general de la plataforma.
Objetivo	Mantener una coherencia entre los objetivos del aprendizaje y los recursos creados.	Realizar un diseño sustentable y pertinente que satisfaga las necesidades de los usuarios de la plataforma de tecnología educativa.
Evaluación	Evaluaciones continuas, sumativas y formativas.	Evaluaciones heurísticas, pruebas de calidad, encuestas de satisfacción.
Calidad	Determinada por la adquisición de conocimiento y la formación de los estudiantes.	Determinada por la percepción de utilidad, funcionalidad, usabilidad y experiencia de usuario.
Proceso	Iterativo, con base en los modelos y principios de diseño instruccional.	Iterativo, con base en modelos y metodologías de desarrollo de software.

Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Coomans & Lacerda (2015) y Mohammadi (2015).

Aunque son evidentes las diferencias entre ambos aspectos, es posible identificar puntos de coincidencia entre estos. Por lo tanto, resulta natural pensar en la necesidad de unificar ambos aspectos, con la finalidad de diseñar y desarrollar un producto práctico y útil en el proceso de enseñanza-aprendizaje, tanto para estudiantes como para docentes (Koehler et al., 2015; Waight et al., 2015).

4. Planteamiento teórico

4.1. Hipótesis

La aplicación de una metodología de desarrollo de software específica para la creación de una plataforma de tecnología educativa facilitará la implementación de modelos de diseño instruccional en dichas plataformas.

4.2. Preguntas de investigación

- ¿Qué efecto tiene la ausencia de un sustento pedagógico en las plataformas de tecnología educativa en el proceso de enseñanza - aprendizaje?
- ¿Cuál es el estado del conocimiento referente al desarrollo de plataformas de tecnología educativa?
- ¿Qué características toman en cuenta los docentes para seleccionar la plataforma de tecnología educativa que han de utilizar para impartir sus clases?

4.3. Objetivos

4.3.1. Objetivo general

Diseñar una metodología de desarrollo de software para la creación de plataformas de tecnología educativa haciendo uso de patrones arquitectónicos y de diseño de software que facilite la aplicación de modelos de diseño instruccional en dichas plataformas.

4.3.2. Objetivos específicos

- Identificar el estado del conocimiento consolidado sobre metodologías de desarrollo de software para la creación de plataformas de tecnología educativa haciendo uso de patrones arquitectónicos y de diseño de software a través de la revisión sistemática de la literatura especializada.
- Explicar la importancia de los patrones arquitectónicos y de diseño de software en la tecnología educativa a través de una cartografía conceptual.
- Aplicar un modelo que unifique los patrones de diseño instruccional con los patrones de diseño de software.

- Desarrollar un instrumento para la evaluación de la percepción de una plataforma de tecnología educativa

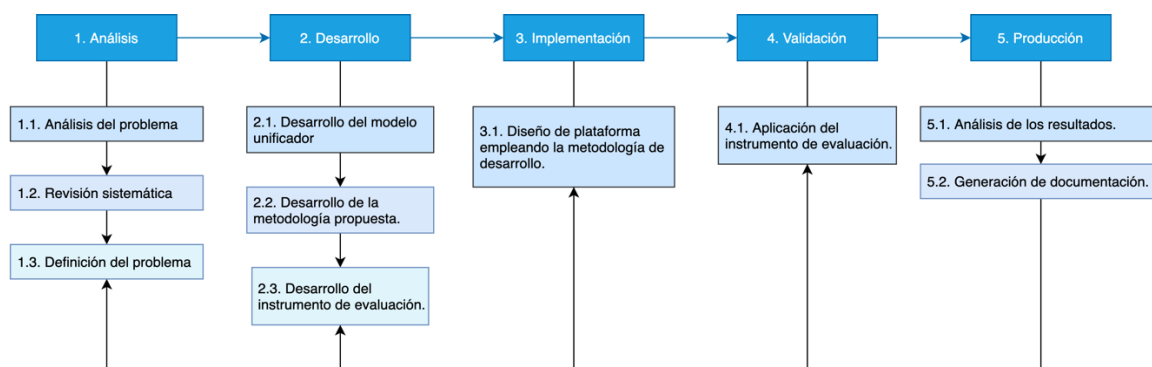
5. Metodología

Para el adecuado desarrollo del proyecto, cumplimiento de los objetivos y comprobación de la hipótesis planteada, se optó por utilizar la metodología de Investigación Basada en el Diseño, por la orientación hacia la innovación educativa y la flexibilidad que ofrece (Benito & Salinas, 2016). Se trata de una metodología sistemática y flexible que hace uso del análisis, diseño, desarrollo e implementación iterativos para una mejora continua (Figura 5.1).

Establecida la ruta crítica, de acuerdo con la Investigación Basada en el Diseño, es indispensable mencionar que, para diferentes aspectos de la misma, será necesario aplicar métodos de probabilidad y estadística (Groth, 2017).

Figura 5.1.

Metodología de investigación



Nota. Elaboración propia con base en Benito & Salinas (2016)

5.1. Análisis conceptual de la temática de investigación

En la primera etapa, destinada al análisis de la situación que estudia el proyecto de investigación, fue indispensable realizar una descripción previa del estado del conocimiento de la temática que rodea al tema de investigación. Según Sánchez-Meca (2010), en esta es necesario realizar una clasificación de variables discretas para efectuar un análisis estadístico que consistirá en describir las características de los estudios que se utilizaron, para después hacer uso de algún modelo estadístico que será elegido en función de la heterogeneidad de los datos obtenidos.

Para este proyecto de investigación, sin embargo, se utilizará el método de revisión sistemática conocido como metasíntesis, el cual permite realizar una síntesis cualitativa de lo descrito por los principales exponentes de un tema de investigación en particular (Salinas & Marín, 2019). Lo anterior, debido a que tanto la tecnología educativa como los procesos de aprendizaje y el diseño instruccional se tratan de conceptos inherentemente cualitativos.

Al tratarse de un tema de investigación novedoso y poco estudiado (como se pudo verificar en la identificación de la problemática), fue necesario emplear estrategias inusuales para la recopilación de documentos. Esto, sin embargo, se hizo manteniendo la calidad de la revisión sistemática, ya que la metasíntesis se fundamenta en la triangulación de expertos del tema, contextos y fuentes de información, donde se busca la replicabilidad y la trazabilidad del proceso (Kugley et al., 2017; Leary & Walker, 2018; Thomas & Harden, 2008).

Así pues, la metasíntesis fue realizada y sus resultados fueron posteriormente publicados. Dentro de esta, se estableció una pregunta guía, ¿cómo se han considerado los principios de Ingeniería de Software en el diseño y desarrollo de plataformas de tecnología educativa?, esto con la finalidad de identificar la barrera del conocimiento, además de poder identificar posibles puntos de coincidencia entre la Ingeniería de Software y el Diseño Instruccional.

Entonces, fue posible responder la pregunta guía, al establecer 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. Además de lo anterior, resultaría complejo (más no imposible) pensar en algún modelo que permitiera contemplar tanto el sustento pedagógico como los principios del proceso de diseño y desarrollo de software. Sin embargo, también fue posible identificar que esta unión podría ser factible en puntos

de coincidencia que ambas disciplinas poseen, los patrones de diseño instruccional y los patrones de diseño de software (Ibarra-Corona & Escudero-Nahón, 2021).

5.2. Desarrollo de la propuesta metodológica

En la etapa de desarrollo, se llevó a cabo la unificación de los patrones de diseño instruccional con los patrones arquitectónicos y de diseño de software en un modelo que servirá como base para el diseño de la metodología de desarrollo de software que busca proponer el proyecto de investigación y que estaría construida con base en los patrones de diseño instruccional y de diseño de software. Dicha unificación se realizó a partir del análisis de la literatura más reconocida en el tema, misma que conforma el marco teórico de la presente investigación y a partir de la cual, fue realizada una cartografía conceptual.

5.3. Implementación de la propuesta metodológica

Una vez desarrollada la metodología, esta fue implementada mediante el diseño de las interfaces de una plataforma de tecnología educativa en un ambiente educativo de nivel superior para tener un producto a través del cual fuera posible verificar la fiabilidad de la metodología propuesta. El proceso de desarrollo de las interfaces involucró un proceso de desarrollo de software, mismo que se encuentra presente dentro de la metodología propuesta.

5.4. Validación de la propuesta

Terminada la etapa de implementación, fue imperativo validar los productos obtenidos (modelo, diseño y metodología). Se diseñó una entrevista como instrumento de evaluación cualitativa para docentes que permita obtener información referente a las capacidades y percepción, tanto de la plataforma diseñada aplicando la metodología, como de la metodología misma (Palacios Vicario et al., 2013). Los resultados de la entrevista fueron posteriormente sintetizados y analizados mediante una codificación axial. De igual forma, se utilizó un instrumento de evaluación cuantitativo previamente probado para contar con datos discretos que complementen los resultados cualitativos (Chen et al., 2017).

Tras la aplicación del instrumento se realizó el análisis y síntesis pertinente de la información obtenida. Con los resultados obtenidos, fue posible realizar las correcciones pertinentes en cada uno de los procesos de las etapas anteriores, y así efectuar una siguiente iteración para obtener el mejor producto final posible tanto en la metodología como en la plataforma diseñada.

5.5. Producción del material suplementario

Obtenidos los resultados deseados, estos serán debidamente documentados en el proyecto de investigación, así como en manuales y documentos diversos donde se verán plasmados los resultados de las comprobaciones realizadas a través de los instrumentos de evaluación. Este material suplementario estará dirigido a docentes.

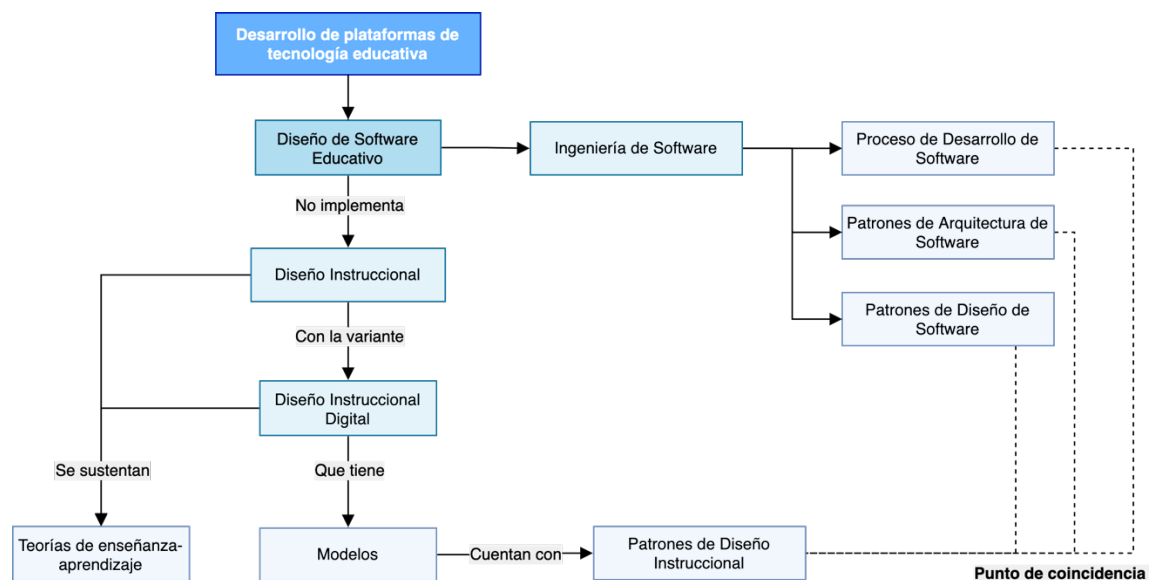
6. Propuesta

6.1. Modelo unificador

Para realizar un modelo que permita unificar la teoría pedagógica con la práctica tecnológica (Chimalakonda & Nori, 2018; Lowyck, 2014), es necesario identificar uno o varios puntos de coincidencia entre ambas disciplinas. Para lograr esto, a partir de la revisión sistemática, se realizó una delimitación conceptual y semántica de los conceptos y componentes que circundan las dos disciplinas principales de esta investigación: la tecnología educativa y la ingeniería de software. En la Figura 6.1, se presenta la delimitación conceptual de ambos conceptos.

Figura 6.1.

Delimitación conceptual de la tecnología educativa y la ingeniería de software



Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Chimalakonda & Nori (2018), Gros et al. (2016) y Seoane & García (2014).

A partir de esta delimitación es posible identificar que, en efecto, existen puntos de coincidencia entre ambas disciplinas. En específico, es posible identificar que las áreas de diseño de ambas áreas (diseño instruccional, diseño de software y diseño arquitectónico) cuentan con patrones (Gros et al., 2016). Estos patrones pueden ser homologados y tomados como punto de partida para la elaboración de un modelo de referencia (Adnan & Ritzhaupt, 2018; Seoane & García, 2014).

No obstante, esta homologación resulta compleja y confusa de sistematizar, puesto que los patrones pedagógicos se sustentan en la aplicación práctica de las teorías de enseñanza y de aprendizaje, aplicación que no se puede formalizar ni sistematizar del mismo modo que un código de programación (Fourcade et al., 2017; Gros et al., 2016).

Así pues, para realizar la homologación de los conceptos principales, es necesario clasificarlos en dos categorías: lo que tenga que ver con lo estructural (arquitectura) y lo que sea referente a algo funcional (diseño) (Koehler & Mishra, 2005). En la Tabla 6.1, se presenta la clasificación realizada.

Tabla 6.1.

Clasificación categórica de los conceptos según su área

Área	Estructural (arquitectura)	Funcional (diseño)
	- Preparación de cursos	
	- Planeación de unidades	
	- Planeación de la instrucción	
Diseño instruccional	- Planeación de la evaluación	-
	- Estrategias de comunicación	
	- Ecosistema tecnológico	
	- Escenarios de enseñanza	
		- Estrategias tecnológicas
		- Estrategias didácticas
		- Competencias
		- Evaluación
Enseñanza-aprendizaje	-	- Comunicación
		- Mediación
		- Reflexión
		- Motivación

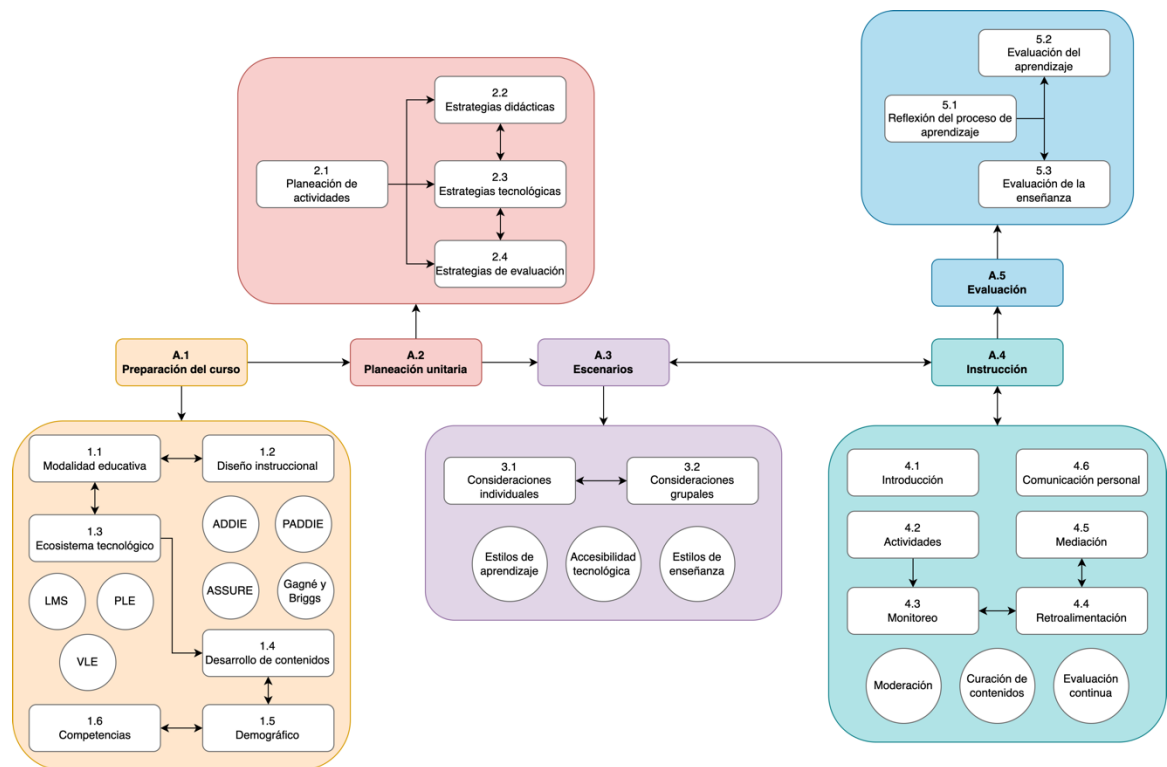
Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Belloch (2017), Koehler & Mishra (2005) y Nori et al. (2014).

La clasificación permitió identificar que los conceptos relacionados con el diseño instruccional son de corte estructural, pues se tratan de conceptos ligados a la *arquitectura* de un proceso de enseñanza-aprendizaje. En pocas palabras, el

diseño instruccional brinda estructura a este proceso (Belloch, 2017). Por su parte, los conceptos relacionados con el proceso de enseñanza-aprendizaje están orientados a cuestiones funcionales y de aplicación de teorías del aprendizaje (Meneses, 2007).

Es así como, a partir de los puntos de coincidencia y de la clasificación conceptual, fue posible plantear un modelo que unifica ambas disciplinas (lo pedagógico con lo tecnológico). En la Figura 6.2, se presenta la estructura del modelo.

Figura 6.2.
Modelo unificador



Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Belloch (2017), Chimalakonda & Nori (2018), Fourcade et al. (2017), Gros et al. (2016), Ritzhaupt & Kumar (2015), y Seoane & García (2014).

El modelo se ordena según cinco patrones estructurales (arquitectónicos), en cada patrón se establecen las actividades de planeación y diseño indispensables

para el desarrollo de un curso en plataformas de tecnología educativa. No obstante, es importante realizar una serie de precisiones sobre este modelo.

En primer lugar, se trata de un modelo de referencia, el cual únicamente establece lo que se debe tomar en consideración, mas no el cómo desarrollar las actividades que plantea. En segundo lugar, es necesario considerar que en este modelo ya se aplican los patrones de diseño y de arquitectura de software como principio y como sustento (por esto, la separación entre lo estructural y lo funcional), mas no de forma técnica pues esto resultaría imposible (Adnan & Ritzhaupt, 2018; Seoane & García, 2014).

Por último, resaltar que cada aspecto estructural del modelo podría tratarse de forma individual, sin que esto afecte su utilidad. Lo anterior, debido a que una de las propiedades de los patrones de arquitectura y de diseño es su modularidad y su independencia de otros patrones (Chimalakonda & Nori, 2018; Navarro et al., 2017). En la Tabla 6.2 se describen a detalle los patrones estructurales y funcionales utilizados en el modelo.

Tabla 6.2.

Definición de los patrones estructurales y funcionales

Patrón	Definición
A.1. Preparación del curso	En este elemento de la arquitectura caben todos los patrones de diseño necesarios para permitir los aspectos estratégicos del proceso de enseñanza-aprendizaje.
1.1. Modalidad educativa	Se definen claramente reglas para los escenarios que se van a afrontar, así como la modalidad de enseñanza que se utilizará.
1.2. Diseño instruccional	Se define un diseño instructivo general con base en las características del curso a desarrollar, este servirá como estructura a la que todas las acciones formativas deben aproximarse (ADDIE, PADDIE, ASSURE, etc.).
1.3. Ecosistema tecnológico	Se toman las decisiones estratégicas sobre el uso de las herramientas necesarias para desarrollar el modelo de <i>e-learning</i> descrito.

Patrón	Definición
1.4. Desarrollo de contenidos	Se define si es necesario dotar a los contenidos de una estructura estándar y, en caso de ser necesario, se crean los modelos necesarios para su producción.
1.5. Demográfico	Se definen las estrategias e instrumentos para recopilar información demográfica sobre los estudiantes, misma que será de utilidad para el diseño de contenidos.
1.6. Competencias	Se definen las estrategias e instrumentos para el análisis de las competencias.
A.2. Planeación unitaria	Se definen los elementos para la planificación de la intervención formativa o diseño instructivo de las actividades formativas concretas de cada unidad temática a enseñar.
2.1. Planeación de actividades	Se define la forma de trabajo y desarrollo de actividades por parte de los estudiantes, para que pueda ser aplicado en las unidades.
2.2. Estrategias didácticas	Se describen las estrategias didácticas según el tipo de competencias.
2.3. Estrategias tecnológicas	Se definen la forma y tiempos en los que se utilizarán las herramientas que se necesitarán para el desarrollo de las actividades.
2.4. Estrategias de evaluación	Definición de las principales estrategias de evaluación que se aplicarán durante el desarrollo del curso.
A.3. Escenarios	Este elemento de la arquitectura define las pautas de actuación ante situaciones en las que es necesario realizar ajustes no previstos en la estrategia de aprendizaje.
3.1. Individuales	Conjunto de estrategias por parte del docente para la resolución de problemas de estudiantes concretos.
3.2. Grupales	Conjunto de estrategias por parte del docente para la resolución de problemas en grupos de estudiantes.
A.4. Instrucción	Definición de las funciones de los docentes. Los elementos de diseño que se incluyen en este componente de la arquitectura se consideran funcionales a una visión del <i>e-learning</i> .

Patrón	Definición
4.1. Introducción	Definición de pautas precisas para el desarrollo de las actividades y del curso.
4.2. Actividades	Modelo para la distribución de tareas, que pueden ir desde el desarrollo de actividades individuales hasta diferentes dinámicas de trabajo en grupo.
4.3. Monitoreo	Estrategias para el seguimiento puntual de las actividades.
4.4. Retroalimentación	Definición de las estrategias de retroalimentación por parte del docente hacia las actividades desarrolladas por los estudiantes.
4.5. Mediación	Estrategias para la mediación de situaciones de conflicto entre pares como en las relaciones con docentes y otros perfiles involucrados en la formación.
4.6. Comunicación personal	Estrategia para la gestión de la comunicación individual y colectiva, así como de las herramientas a utilizar en cada caso y el modo en que se utilizan.
A.5. Evaluación	Definición de las estrategias de evaluación y medición de la calidad del proceso de instrucción.
5.1. Reflexión	Proceso de reflexión para la mejora de calidad de todo el proceso formativo.
5.2. Aprendizaje	Definición de las estrategias y herramientas para la evaluación del aprendizaje de los estudiantes durante el proceso de instrucción.
5.3. Enseñanza	Definición de las estrategias y herramientas para la evaluación del desempeño del docente durante el proceso de instrucción.

Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Belloch (2017), Chimalakonda & Nori (2018), Fourcade et al. (2017), Gros et al. (2016), Ritzhaupt & Kumar (2015), y Seoane & García (2014).

Así pues, el modelo se compone de cinco patrones de arquitectura (estructurales), dentro de los cuales, se encuentran 21 patrones de diseño. Los patrones estructurales definen de manera general la estructura que se debe seguir

en el diseño y desarrollo de cualquier proceso de enseñanza-aprendizaje mediado por la tecnología. Por su parte, los patrones de diseño especifican los aspectos relativos a la operatividad y desarrollo de un proceso de enseñanza-aprendizaje, como pueden ser: los escenarios, indicadores, estrategias, herramientas, técnicas y tecnologías que el docente utilizará.

6.2. Metodología propuesta

Para la elaboración de una metodología para el desarrollo de plataformas de tecnología educativa con base en patrones arquitectónicos y de diseño de software, es necesario tener algunas consideraciones. Para empezar, la ingeniería de software ya cuenta con múltiples metodologías para el desarrollo de software, estas se encuentran orientadas a la transformación de necesidades (requerimientos) en soluciones, a la vez que se asegura la calidad de las mismas mediante el uso de métricas y estándares (Oyelere et al., 2018).

Lo anterior, obligó a la presente investigación a analizar y comparar las metodologías existentes. La finalidad de esta comparativa fue identificar aquellos elementos que, en conjunto con el modelo planteado con anterioridad, permitan realizar una propuesta metodológica viable y capaz de cumplir no solo con los objetivos planteados, sino que también pueda poner a prueba la hipótesis de la investigación. Para realizar este análisis y comparación de las metodologías, se utilizó el método comparativo (Gómez Díaz de León, 2014), caracterizado por ser utilizado en investigaciones cualitativas de carácter social.

Este método se desarrolló en tres etapas donde, en la primera, se delimitaron de forma conceptual y semántica aquellas metodologías propias de la ingeniería de software, tal y como se presentó en el marco teórico. En una segunda etapa, se realizó una selección muestral de las metodologías existentes. Mientras que, en una tercera etapa, se realizó el análisis comparativo con base en cuatro indicadores pertinentes para la investigación: tipo de desarrollo, flexibilidad, curva de aprendizaje e involucramiento del usuario. Estos hacen referencia a cualidades de las metodologías e indican el nivel técnico de conocimientos requerido para su uso.

Las metodologías, como se había mencionado con anterioridad, se clasifican en tradicionales y ágiles. Se optó por seleccionar únicamente algunas de las metodologías ágiles para su comparación, debido a ciertas características y atributos en común con el diseño instruccional como lo son: flexibilidad, agilidad, curva de aprendizaje e involucramiento del usuario (Özcan-Top & Demirors, 2019; Papadopoulos, 2015). Debido a lo anterior, se seleccionaron a las metodologías *Crystal*, *DSDM*, *Kanban*, *Scrum* y *XP* para ser analizadas y comparadas. En la Tabla 6.3 se presentan los resultados de la comparación.

Tabla 6.3.

Análisis comparativo de las metodologías

Metodología	Tipo de desarrollo	Flexibilidad	Curva de aprendizaje	Involucramiento del usuario
Crystal	Iterativo	Alta	Normal	Alto
DSDM	Iterativo	Baja	Difícil	Bajo
Kanban	Iterativo	Media	Normal	Bajo
Scrum	Iterativo	Alta	Normal	Alto
XP	Iterativo	Baja	Fácil	Alto

Nota. Elaboración propia con base en Navarro-Cadavid (2013).

Es posible apreciar que, incluso dentro de una misma clasificación (ágiles), cada metodología tiene similitudes y diferencias cuando son comparadas en los cuatro rubros utilizados. Para los propósitos de la presente investigación, las características que mayor peso tienen son: flexibilidad, involucramiento del usuario y la curva de aprendizaje. Lo anterior, debido a que se pretende que la metodología sea lo más adecuada posible para el desarrollo de plataformas de tecnología educativa, para lo cual, involucrar a los docentes, investigadores y alumnos es indispensable (Nori et al., 2014).

A partir de este análisis comparativo, fue posible identificar las metodologías *Scrum* y *Crystal* como las que mejor se adaptarían a desarrollos del ámbito educativo, debido a que cuentan con características que les permitirían ser

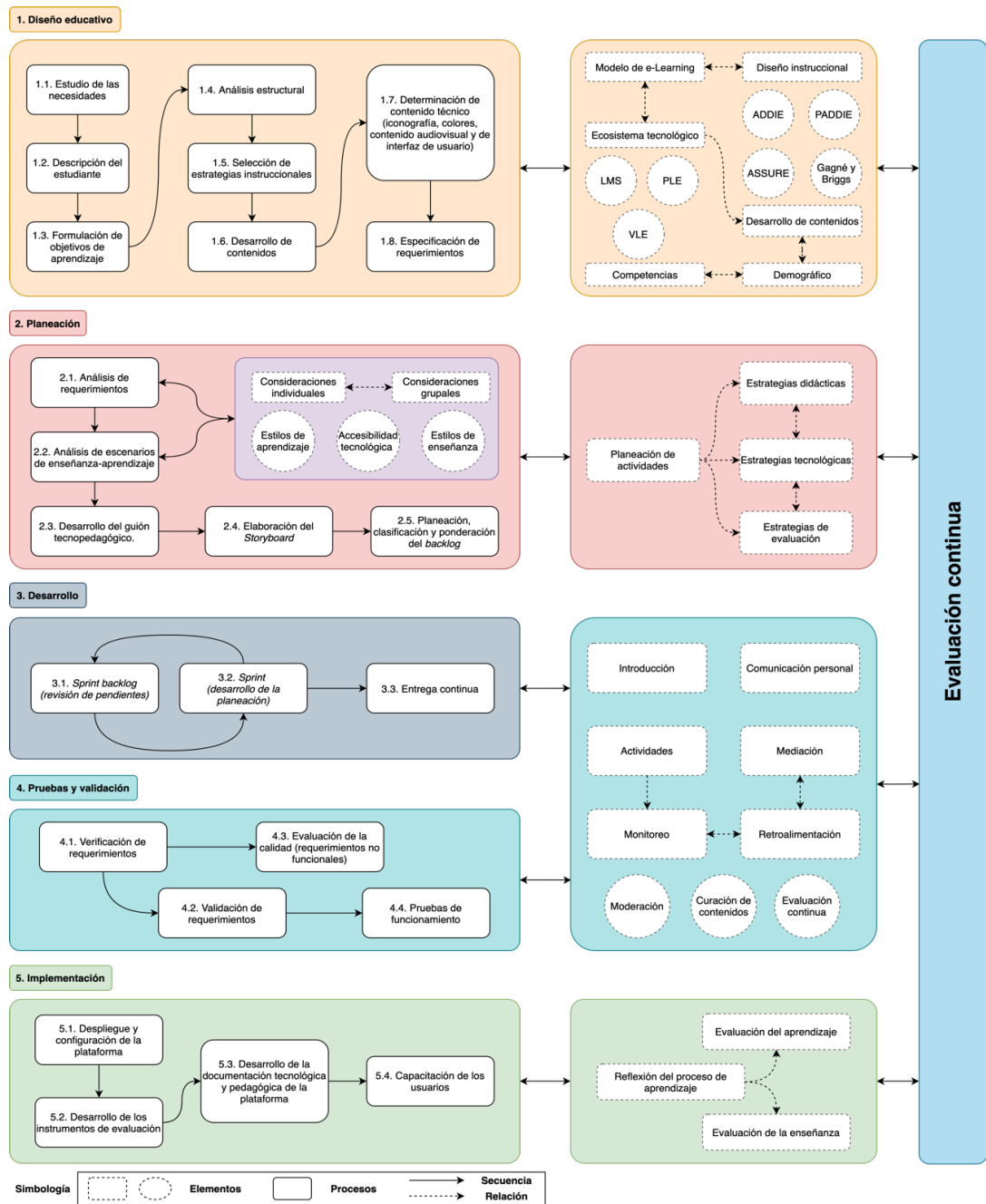
modificadas y adecuadas a un proceso de diseño instruccional. Estas metodologías, al ser pensadas como metodologías de propósito general, son flexibles. Además, cuentan con una curva de aprendizaje no tan pronunciada lo que facilita su uso (es decir, se pueden aprender a utilizar rápidamente), comprensión y aplicación por personas que no cuenten necesariamente con los conocimientos técnicos o con la formación propia de un ingeniero de software (Dybå & Dingsøyr, 2008).

Por una parte, *Scrum* se trata de una de las metodologías más populares en la actualidad (Özcan-Top & Demirors, 2019), lo que a su vez le permite ser la más documentada, utilizada y conocida. Por otra parte, *Crystal* se trata de una metodología abierta y libre, lo que la convierte en una metodología ágil y flexible, aunque también la convierte en una de las más complicadas de comprender, entender y aplicar, ya que depende mucho de la experiencia de quien la emplea y, por lo tanto, no es tan fácil de unir con otras metodologías (Jabangwe et al., 2018).

Es notorio que, si bien estas metodologías podrían ser empleadas para el diseño y desarrollo de plataformas de tecnología educativa, tienen carencias cuando se trata de trasladarlas a este dominio. De inicio, aunque se presentan como metodologías con un alto grado de flexibilidad, lo cierto es que una metodología entre más flexible es, menos estructura y apoyo da a aquellos que deciden utilizarla, lo cual resultaría contraproducente, ya que se busca proponer una metodología en la que el nivel de experiencia no sea un factor determinante (Jabangwe et al., 2018).

De igual forma, aunque en las metodologías de desarrollo de software se propone que no es necesario ser un experto, la realidad es que estas utilizan términos y conceptos técnicos que pertenecen a la ingeniería de software y que no cualquiera conoce. Así pues, con base en los procesos establecidos en *Scrum* y en *Crystal*, así como con lo establecido en el modelo unificador, es posible plantear una metodología adecuada para el desarrollo de plataformas de tecnología educativa que considere no solo los aspectos técnicos, sino la parte pedagógica del proceso de enseñanza-aprendizaje mediado por tecnologías (Rodríguez, 2014). En la Figura 6.3 se presenta la metodología propuesta, titulada PADSE.

Figura 6.3.
Metodología PADSE



Nota. PADSE es el acrónimo para Patrones de Arquitectura y Diseño de Software Educativo. Elaboración propia con base en Adnan & Ritzhaupt (2018), Chimalakonda & Nori (2018), Coomans & Lacerda (2015), Fourcade et al. (2017), Reyes (2007), Ritzhaupt & Kumar (2015), Rodríguez (2014) y Seoane & García (2014).

La metodología PADSE se compone de cinco etapas, dentro de las cuales se desarrollan 24 procesos encaminados a cumplir con los aspectos técnicos, tecnológicos y pedagógicos que una plataforma educativa debe considerar. En la metodología es posible apreciar la implementación del *modelo unificador* (Figura 6.2). En la Tabla 6.4 se describen de forma general las etapas y procesos comprendidos en la metodología propuesta.

Tabla 6.4.

Descripción de las etapas y procesos de la metodología propuesta

Etapas y procesos	Descripción
1. Diseño educativo	Esta primera etapa tiene como propósito el diseño y planteamiento de todo lo relativo al proceso de enseñanza-aprendizaje dentro de una plataforma de tecnología educativa. Consta de ocho diferentes procesos.
1.1. Estudio de las necesidades	Primer proceso de la primera etapa. Se determinan las necesidades y características del curso, tales como el tiempo a emplear en una actividad o clase, la cantidad de contenido, la cantidad de alumnos, automatización de procesos que no interesan como contenido, generación de actividades de refuerzo, etc.
1.2. Descripción del estudiante	Se define cuál es el público potencial para seleccionar aspectos relacionados con la cultura, costumbres, preferencias, edades, estilos de aprendizajes, etc.
1.3. Formulación de los objetivos de aprendizaje	En esta parte se redactan los objetivos generales y específicos que se quieran alcanzar con el uso de la plataforma.
1.4. Análisis estructural	Se especifican las habilidades a desarrollar, se toman en cuenta los atributos básicos de los conceptos que se quieran trabajar y cómo se tendrían que ver estos plasmados en la plataforma.
1.5. Selección de estrategias instruccionales	Se definen los eventos de aprendizaje que sean considerados necesarios por el docente para lograr los objetivos establecidos. Se determina la mejor manera en que un determinado contenido va a ser presentado al estudiante. Es necesario hacer una revisión de los modelos de diseño instruccional con mayor afinidad al curso a impartir para poder definir las acciones a seguir. El docente tiene que pensar que está diseñando un curso para implementarlo en un medio que no es estático, sino dinámico.
1.6. Desarrollo de contenidos	Aquí se selecciona y organiza a detalle el contenido temático que se tratará en la plataforma. Se hace una lista de temas, recursos y herramientas complementarias que tendrán que integrarse en la misma.

Etapas y procesos	Descripción
1.7. Determinación de contenido técnico	En este proceso se especifican aspectos relacionados con aspectos de diseño, principio de orientación, uso de íconos, botones, fondos, textos, contenido multimedia, colores, prestaciones (como concurrencia y capacidad de respuesta), etc.
1.8. Especificación de los requerimientos	Realizado todo lo anterior, todo se tiene que plasmar en un documento de especificación de requerimientos (SRS, por sus siglas en inglés). En este documento se organiza lo planteado y diseñado en los procesos anteriores y se estructura para su desarrollo.
2. Planeación	En esta segunda etapa se pretende que, a partir del SRS generado en la primera etapa, se realice todo el proceso de planeación para continuar a la etapa de desarrollo.
2.1. Análisis de los requerimientos	Aquí, los requerimientos plasmados en el SRS son analizados y categorizados según el nivel de importancia y urgencia que tengan dentro del desarrollo final del sistema.
2.2. Análisis de escenarios de enseñanza-aprendizaje	En este proceso, se identifican dentro de los requerimientos los posibles escenarios en los que se desarrollará la plataforma. Se toman en consideración aspectos generales e individuales de los estudiantes y docentes, tales como: competencia y accesibilidad tecnológica, estilos de enseñanza y estilos de aprendizaje.
2.3. Desarrollo del guion tecnopedagógico	Con base en lo anterior y en las características ya definidas y analizadas de la plataforma, se desarrolla un guion tecnopedagógico en el que se plantea el uso de los recursos de la plataforma y los complementarios encaminados a cumplir los objetivos de aprendizaje.
2.4. Elaboración del <i>storyboard</i>	Es el resultado de la visualización del guion tecnopedagógico, en el que se integran el desarrollo mediado por tecnología de: contenidos, elementos multimedia, tiempos de enseñanza y evaluación.
2.5. Planeación, clasificación y ponderación del <i>product backlog</i>	Como producto de esta etapa, se realiza el <i>backlog</i> de la plataforma a desarrollar. El <i>backlog</i> es una lista de trabajo ordenado por prioridades, que se obtiene del SRS y que simplifica la planificación de iteraciones y entregas continuas.
3. Desarrollo	En esta etapa, se elabora el aspecto más técnico de la metodología, el cual consiste en el desarrollo del software mediante el uso de tecnologías existentes (como puede ser el uso de ambientes virtuales de enseñanza existentes) o mediante el uso de lenguajes de programación. Esta etapa se realiza de forma iterativa e incremental (múltiples veces con entregas constantes).

Etapas y procesos	Descripción
3.1. <i>Sprint backlog</i>	Este proceso consiste en realizar en cada iteración un plan de acción para el desarrollo de aspectos incrementales del producto. Para cada iteración, se seleccionan elementos del <i>product backlog</i> definido en la etapa anterior.
3.2. <i>Sprint</i>	Aquí, se ejecuta la planeación de cada iteración del proceso anterior, con el objetivo de realizar un incremento en el desarrollo funcional del proyecto de software. En este caso, de la plataforma de tecnología educativa.
3.3. Entrega continua	Se generan productos incrementales del proyecto definido en etapas anteriores. Estos productos, son integrados con todos los otros elementos desarrollados en cada iteración.
4. Pruebas y validación	Esta etapa tiene por objetivo, comprobar, validar y retroalimentar los componentes desarrollados en la etapa anterior, se verifica y comprueba que se apeguen al diseño y planeación realizados en las etapas 1 y 2 respectivamente.
4.1. Verificación de requerimientos	En este proceso, se verifica que todos los requerimientos planteados fueron desarrollados y se apegan al diseño y planeación realizados.
4.2. Validación de requerimientos	Este proceso consiste en asegurar que los requisitos desarrollados en la etapa anterior reflejan las expectativas de los interesados.
4.3. Evaluación de la calidad	Consiste en realizar una evaluación de la calidad de un producto de software concreto según las características suplementarias que se hayan definido en virtud de las necesidades del proyecto. Esta evaluación se realiza mediante listas de cotejo.
4.4. Pruebas de funcionamiento	La realización de las pruebas consiste en la ejecución, revisión y retroalimentación de las funcionalidades previamente planeadas y diseñadas para el software. Estas pruebas las realizará el docente de forma conjunta con un grupo de prueba.
5. Implementación	El propósito de esta etapa es el de implementar el producto de la etapa 3, una vez que haya sido validado por la etapa 4. Aquí se realizan las configuraciones pertinentes para la plataforma, así como también se genera la documentación derivada de la misma.
5.1. Despliegue y configuración de la plataforma	Este proceso consiste en la configuración y aprovisionamiento de los entornos en los que se almacenará la plataforma. Esto, en virtud de las necesidades que existan para la misma en términos de accesibilidad, rendimiento y escalabilidad.
5.2. Desarrollo de los instrumentos de evaluación	Los elementos de evaluación, definidos desde el proceso de estrategias instruccionales, se configuran y presentan en la plataforma de tecnología educativa en conjunto con los recursos de esta.

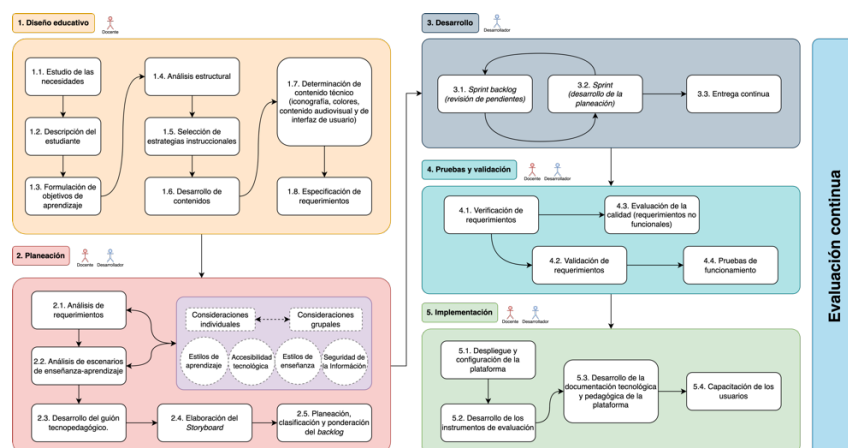
Etapas y procesos	Descripción
5.3. Desarrollo de la documentación tecnológica y pedagógica	Todo lo elaborado en las secciones anteriores, tanto lo referente a los aspectos tecnológicos como los pedagógicos. Esta información apoyará en futuros eventos de mantenimiento y a los usuarios de la plataforma (docentes, alumnos y desarrolladores).
5.4. Capacitación de los usuarios	Con base en la documentación realizada, se capacitan a los usuarios de la plataforma (alumnos). En esta capacitación, se proporciona la documentación realizada en el proceso anterior. La capacitación puede ser realizada por el mismo docente o por los desarrolladores.

Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Chimalakonda & Nori (2018), Fourcade et al. (2017), Reyes (2007), Ritzhaupt & Kumar (2015), Rodríguez (2014) y Seoane & García (2014).

Adicionalmente, cada etapa y proceso de la metodología se encuentra en un estado permanente de evaluación, esto permite que cada etapa pueda retroalimentar e iterar sobre el trabajo realizado en otra etapa diferente, lo que facilita el desarrollo de componentes de software de forma ágil, al mismo tiempo que posibilita la inclusión de cambios y adecuaciones menores según lo dicten los aspectos educativos o técnicos de la plataforma diseñada, esto gracias a la inclusión de los procesos adaptados de *Scrum* y *Crystal* (Budoya et al., 2019). En la Figura 6.4 se presenta una versión sintetizada de la metodología propuesta.

Figura 6.4.

Metodología PADSE sintetizada



Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Chimalakonda & Nori (2018), Coomans & Lacerda (2015), Fourcade et al. (2017), Reyes (2007), Ritzhaupt & Kumar (2015), Rodríguez (2014) y Seoane & García (2014).

7. Resultados

Con la finalidad de comprobar la validez de la metodología, se aplicó una entrevista a siete docentes de diferentes áreas del conocimiento para identificar la utilidad y percepción de las partes pedagógicas y técnicas que componen a la metodología (Piza Burgos et al., 2019; Thelwall & Nevill, 2021). Debido al enfoque tan específico y especializado del tema principal de la investigación, la selección muestral de los entrevistados se realizó de forma deliberada (Chhabra, 2021), bajo los siguientes criterios:

- a) Que tuvieran experiencia en la docencia o en ámbitos educativos.
- b) Que no impartan clases de la misma área del conocimiento que algún otro de los entrevistados.
- c) Que den clases en nivel superior.
- d) Que hubieran estado involucrados previamente en los procesos de configuración, diseño o desarrollo de plataformas o herramientas de tecnología educativa (mismo que se comprobó con una pregunta inicial).

Se trató de una entrevista semiestructurada, a fin de permitir la introducción de preguntas de seguimiento a temas de interés descritos por los entrevistados (Hughes et al., 2022) y constó de ocho preguntas abiertas. Las entrevistas fueron sintetizadas utilizando la plataforma *ATLAS.ti*, donde tras un proceso de codificación axial y análisis de las entrevistas, fue posible clasificar las respuestas de los entrevistados en siete categorías temáticas. Estas categorías temáticas se describen en la Tabla 7.1.

Tabla 7.1.

Categorías temáticas resultantes del análisis cualitativo de las entrevistas

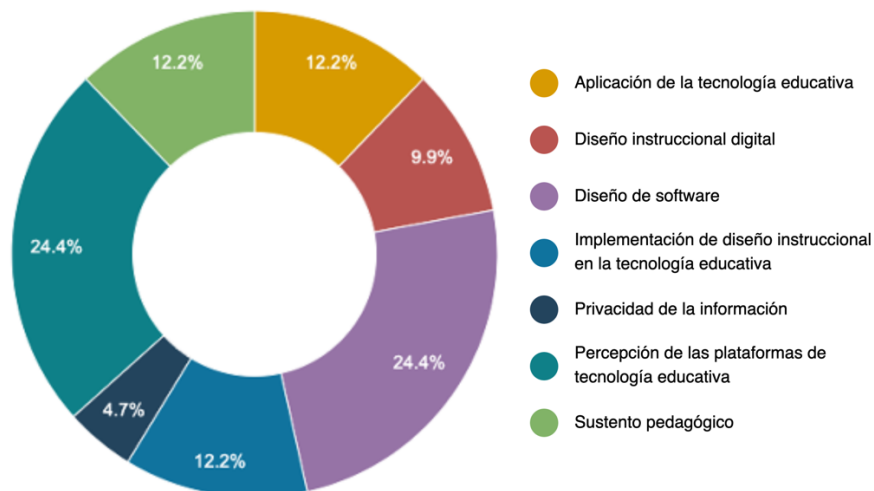
Categoría	Código	Descripción
Percepción de las plataformas de tecnología educativa	PTE	Se refiere a cómo percibe el docente una determinada plataforma de tecnología educativa.

Categoría	Código	Descripción
Aplicación de tecnología educativa	ATE	Forma en la que el docente utiliza la tecnología educativa para el desarrollo de las sesiones de una determinada materia.
Diseño instruccional digital	DID	Nociones de diseño instruccional digital que posee el docente, sean o no aplicadas en su quehacer.
Implementación de diseño instruccional en la tecnología educativa	IDT	Aplicación de las nociones de diseño instruccional en la tecnología educativa empleada por el docente en sus clases.
Sustento pedagógico	SPTE	Sustento pedagógico presente en las plataformas y herramientas de tecnología educativa empleadas por el docente.
Diseño de software	DS	Diseño del software para dar soporte a los aspectos pedagógicos y de patrones arquitectónicos.
Privacidad de la información	PI	Percepción de la seguridad y privacidad de la información por parte del docente en la plataforma y/o herramientas que utilice para sus clases.

La frecuencia relativa con la que estas categorías temáticas fueron identificadas en las respuestas de los entrevistados se presentan en la Figura 7.1.

Figura 7.1.

Frecuencia relativa de las categorías temáticas

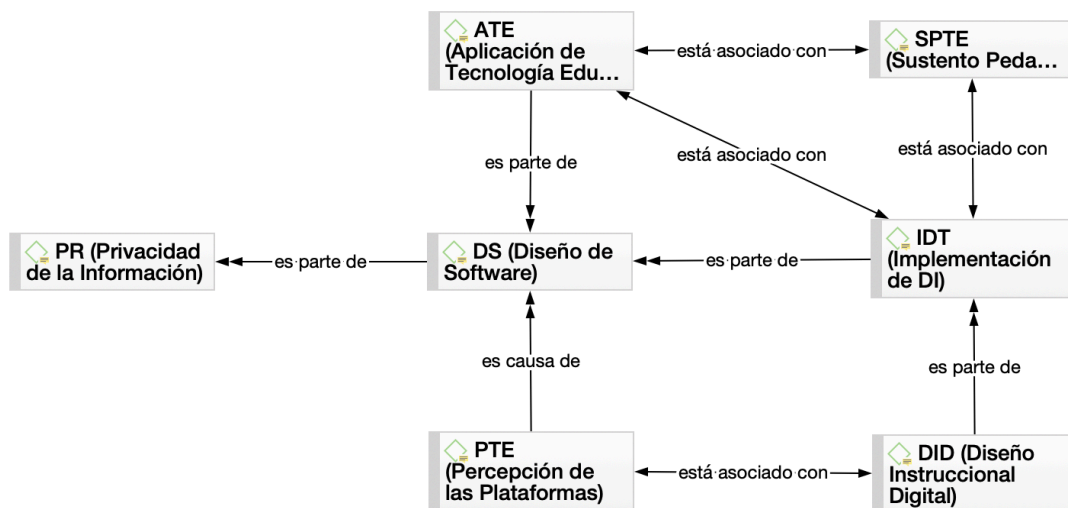


Del análisis de las categorías temáticas, es posible identificar que la privacidad de la información, aunque se trató de la categoría con menor frecuencia relativa y de un apartado que no se contempla en la metodología propuesta, sí fue mencionado por algunos de los entrevistados. Lo anterior implica un interés por la privacidad y la seguridad (Seckler et al., 2015) que no se contempla en la metodología y que es posible añadir. Así también, es posible decir que aspectos de la percepción de la tecnología educativa y del diseño de software fueron comentados en gran proporción de las respuestas de las entrevistas.

No obstante, un análisis o síntesis de tipo cualitativo no se compone únicamente de la revisión de frecuencias, ya que estas pueden significar poco o nada cuando se les compara con lo que en realidad expresan los entrevistados (Palacios Vicario et al., 2013). Esto último, tiene implicaciones respecto a la validez de un análisis cualitativo, ya que es necesario realizar una interpretación hermenéutica de los resultados, en la que se incluyan múltiples fuentes de información (Piza Burgos et al., 2019). Debido a lo anterior, lo primero que se realizó fue una codificación axial de las categorías temáticas (Figura 7.2), con la finalidad de identificar las relaciones existentes entre estas.

Figura 7.2.

Codificación axial de las categorías temáticas



En la codificación axial se presenta como es que las diferentes categorías interactúan entre sí, lo que se muestra en las respuestas de los entrevistados. Donde los mismos hicieron notar que independientemente de su formación, fueron capaces de comprender los aspectos pedagógicos, tecnológicos y de diseño instruccional utilizados en la metodología.

Uno de los hechos más interesantes que se pudieron obtener y que se hizo evidente una vez realizada la codificación axial, fue la falta de la mención de aspectos relacionados con la evaluación, ya que, en las respuestas sobre la metodología, únicamente se mencionaba como un aspecto propio del software y no del proceso de enseñanza-aprendizaje o como componente pedagógico presente en la metodología (Clemmensen, 2004). Esto, podría indicar que se trata de un componente que se tiene que agregar o resaltar en la metodología para su mejor comprensión.

Otro aspecto por resaltar es la mención que hicieron los entrevistados sobre la utilidad de la metodología en dos diferentes vertientes. La primera, hace referencia a la utilidad y capacidad de la metodología para diseñar y crear plataformas, herramientas y tecnologías específicas para una materia o curso en particular, es decir, para desarrollar soluciones *ad hoc* (e.g. Potkonjak et al., 2016). Uno de los aspectos negativos que fueron mencionados con mayor incidencia era que las soluciones de tecnología educativa actuales son genéricas y, por lo mismo, resultan complicadas de utilizar efectivamente como herramientas propias del proceso de enseñanza-aprendizaje. De esta forma se ejemplifica la diferencia entre administrar el aprendizaje y administrar un curso, ya que en el primero sí se considera el proceso de aprendizaje de los estudiantes y los métodos de enseñanza del profesor, mientras que en el segundo no (O'Dell, 2020).

Entonces, al tomar en consideración lo que fue mencionado, en conjunto con las experiencias descritas por los entrevistados, así como lo identificado e interpretado a lo largo del desarrollo de la presente investigación, es posible afirmar que uno de los aspectos que toma mayor relevancia para el profesor en una

plataforma de tecnología educativa es el grado de personalización que esta le ofrece para implementar (según su criterio y conocimiento) el mejor diseño instruccional para un curso en cuestión. En este sentido, la metodología cumple con las expectativas, al partir del diseño del profesor (Adnan & Ritzhaupt, 2018; Bartolomé et al., 2018; M. J. Koehler & Mishra, 2005) para el desarrollo del resto de sus componentes.

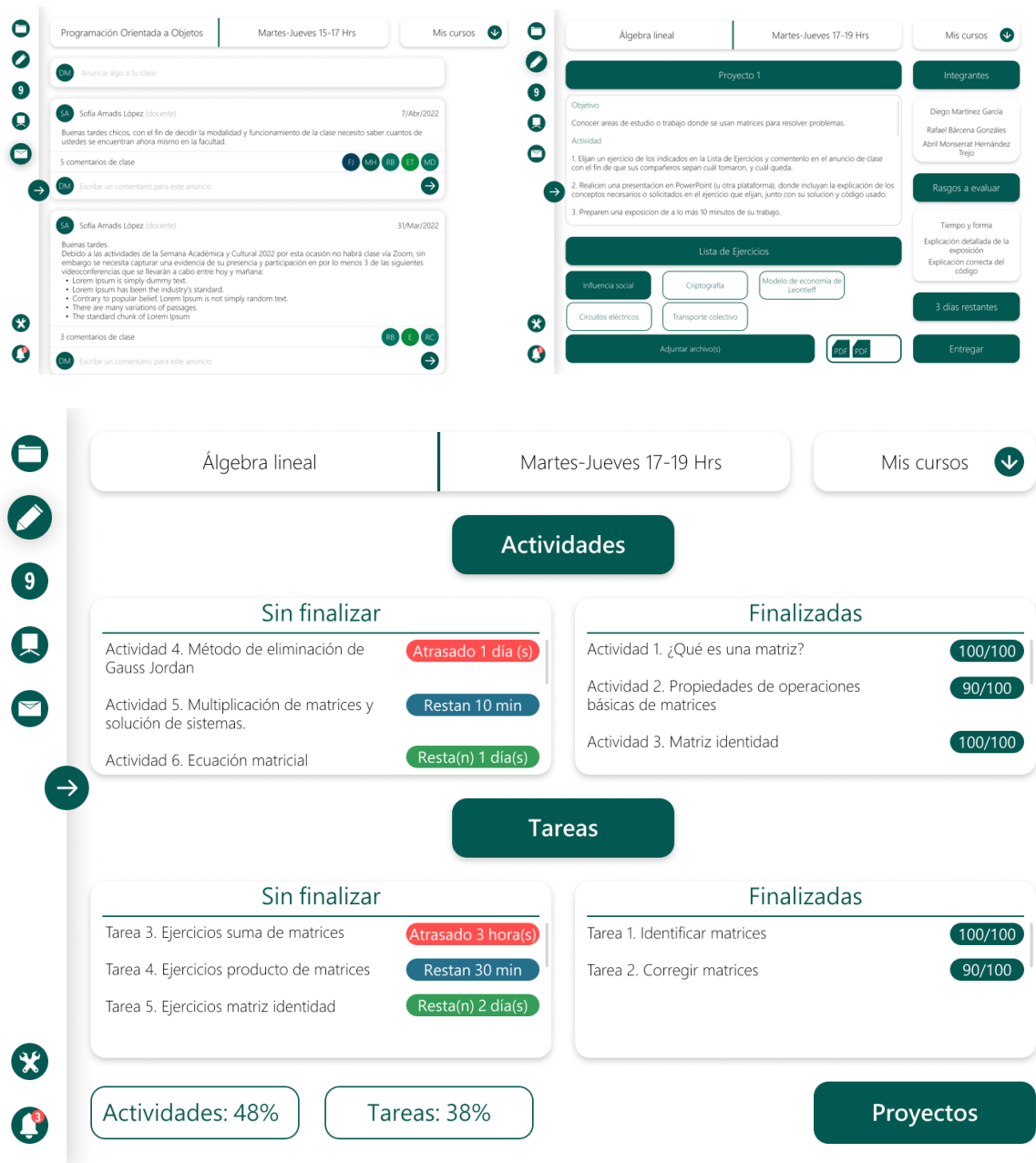
Por otra parte, como segunda vertiente en la que la utilidad de la metodología fue descrita por los entrevistados, se hace referencia a la capacidad de esta de ser utilizada de forma independiente y como auxiliar en la selección de múltiples herramientas de tecnología educativa para un curso. Lo anterior, sin tener que emplear la metodología para el desarrollo de una plataforma de tecnología educativa. Esto, según se mencionó en las respuestas, se debía a que la metodología les permitía comprender los componentes y características que conforman una solución de tecnología educativa (Antonenko et al., 2017). Esto último resulta de interés, al considerar que la mayoría de los entrevistados no pertenecían a un área de formación relacionada con las TIC (debido a los criterios de selección).

Así pues, tanto la metodología como el modelo presentan la posibilidad de emplearse sin utilizar necesariamente lo relativo al proceso de desarrollo de software, sino como referentes para la agregación de tecnologías, herramientas y plataformas de tecnología educativa diversas.

No obstante, no sería adecuado que la metodología únicamente fuera evaluada conceptualmente. Por lo que debería ser aplicada en el proceso de desarrollo de una plataforma para identificar su utilidad en el aspecto tecnológico. Así pues, se realizó el diseño de una plataforma de tecnología educativa, con base en la cual, se evaluaría la efectividad de la metodología. En la Figura 7.3, se presenta el diseño de la plataforma desarrollada.

Figura 7.3.

Interfaces de la plataforma desarrollada con base en la metodología PADSE



La plataforma fue desarrollada considerando todos los aspectos propuestos por la metodología, misma que plantea un proceso continuo de evaluación a lo largo del proceso de planeación, diseño, implementación y validación. Para este proceso de evaluación, se recurrió a una versión modificada del *Instrumento Ergonómico*

Pedagógico para la Evaluación del Software Educativo (PETESE, por sus siglas en inglés), propuesto por Coomans & Lacerda (2015). Este instrumento propone la evaluación de los aspectos tecnológicos, pedagógicos y de usabilidad de un software educativo, mediante seis ejes: (1) aspectos generales, (2) aspectos pedagógicos, (3) aspectos de usabilidad, (4) aspectos técnicos, (5) contenido y (6) interfaz.

Esta versión modificada une los ejes de aspectos de usabilidad (3) y de interfaz (6), al considerarse que son dependientes uno del otro, además de agregar un nuevo eje titulado: experiencia de usuario. Este nuevo eje hace referencia al aspecto cualitativo que todo software posee y que toma en consideración la motivación y sentir del usuario, indispensables en el proceso de enseñanza-aprendizaje (Opel & Rhodes, 2018). Así pues, el instrumento modificado cuenta con la siguiente estructura: (1) aspectos generales, (2) aspectos pedagógicos, (3) aspectos de usabilidad e interfaz de usuario, (4) aspectos técnicos, (5) contenido y (6) experiencia de usuario.

Cada uno de los ejes cuenta con múltiples ítems, que funcionan de forma similar a una lista de cotejo. PETESE está diseñado para ser utilizado por los docentes, desarrolladores y diseñadores instruccionales de la plataforma (Coomans & Lacerda, 2015), al permitir identificar la ausencia de algún elemento necesario antes de implementar la plataforma educativa. En la Tabla 7.2, se presenta el instrumento aplicado a la plataforma desarrollada.

Tabla 7.2.

PETESE aplicado a la plataforma desarrollada

Eje	Ítem	Cumple
1.- Aspectos generales	1.1. El software es innovador y tiene valor agregado	Sí
	1.2. El software es fácil de utilizar	Sí
	1.3. El software tiene un enfoque multidisciplinario	Sí
	1.4. El software tiene niveles diferentes de complejidad	Sí
	1.5. Existe una curva de aprendizaje apropiada para el usuario	Sí
2.- Aspectos pedagógicos	2.1. Existen objetivos educacionales claramente presentados	Sí
	2.2. Es posible utilizar el software de forma individual y colaborativa	Sí

Eje	Ítem	Cumple
	2.3. Promueve la creatividad	Sí
	2.4. Se motiva la exploración y el aprendizaje autodirigido	Sí
	2.5. Se incentiva el autoanálisis y la autorreflexión	Sí
	2.6. Las actividades están planteadas en virtud de los objetivos de aprendizaje	Sí
	2.7. Las actividades son relevantes, auténticas y cuentan con relaciones al mundo real según el contexto del estudiante	Sí
	2.8. El nivel de dificultad para las actividades es el adecuado para el perfil del estudiante	Sí
	2.9. Las actividades son variables	Sí
	2.10. Se presentan los resultados y evaluación de las actividades	Sí
	2.11. Las actividades se pueden personalizar y crear con facilidad	Sí
	2.12. Las tareas están enfocadas a la resolución de problemas y ayudan a desarrollar el pensamiento de orden superior mediante competencias y conceptos	Sí
	2.13. El software puede ser utilizado sin un instructor	Sí
	2.14. El profesor tiene el rol de guía, monitor, tutor y/o facilitador	Sí
	2.15. El software reconoce y apoya el rol del profesor	Sí
	2.16. La evaluación está presente en todos los apartados	No
	2.17. La evaluación es auténtica y está ligada con la enseñanza	Sí
3.- Aspectos de usabilidad e interfaz de usuario	3.1. El sistema guía y alienta al usuario	Sí
	3.2. Hay recursos de ayuda en el sistema	No
	3.3. Las instrucciones son claras y visibles	Sí
	3.4. La retroalimentación es utilizada con efectividad	Sí
	3.5. El software tiene una sección para la discusión y colaboración	Sí
	3.6. El software reconoce errores y se recupera de ellos	Sí
	3.7. El software previene errores del usuario	Sí
	3.8. Los mensajes de error son claros	Sí
	3.9. El usuario puede aprender de sus errores con facilidad	Sí
	3.10. El aprendiz tiene control del sistema y de sus acciones	Sí
	3.11. El software permite la personalización en sus ajustes	No
	3.12. La estructura es sencilla y rápida de comprender	Sí
	3.13. La navegación entre los apartados es coherente	Sí
	3.14. La tipografía es clara y adaptable a diferentes resoluciones	Sí
	3.15. Las acciones que se pueden hacer son explícitas	Sí
	3.16. El diseño es agradable a la vista y es coherente en el software	Sí
	3.17. Los íconos utilizados son representativos	Sí
	3.18. Los colores permiten comprender el software	Sí
4.- Aspectos técnicos	4.1. El software es compatible con el hardware para el que fue diseñado	Sí
	4.2. Las condiciones de compatibilidad son explicadas con claridad	No
	4.3. El software es fácil de instalar	Sí
	4.4. Las modalidades de uso son explicadas	No
	4.5. El software puede manejar grandes cantidades de datos	Sí
	4.6. Los datos generados pueden ser analizados	Sí
	4.7. El software es transparente con el manejo de los datos	Sí

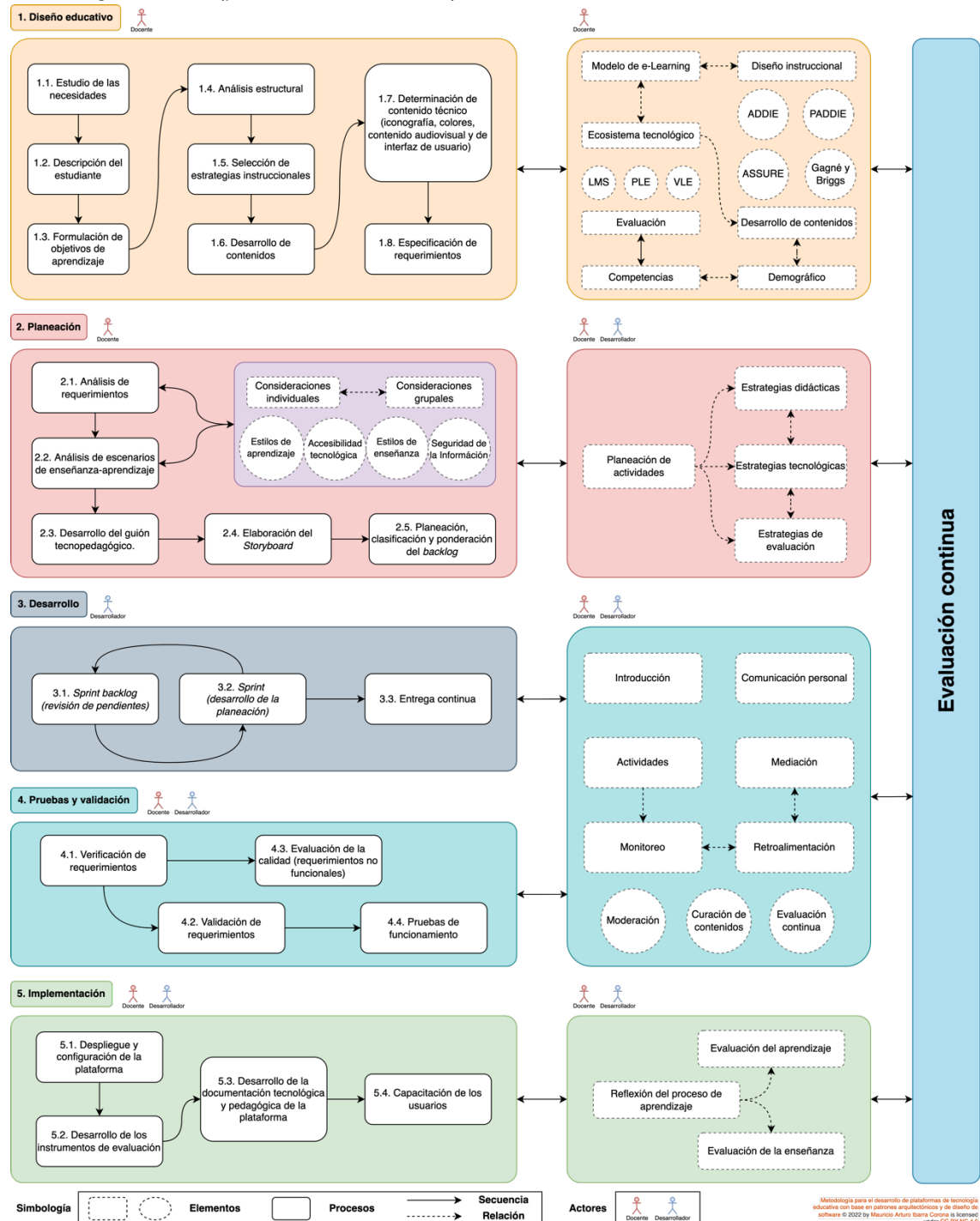
Eje	Ítem	Cumple
	4.8. El usuario almacena la información del usuario	Sí
	4.9. El software será mantenido una vez liberado	Sí
	4.10. El software soportará actualizaciones futuras	Sí
5.- Contenido	5.1. El lenguaje utilizado es adecuado para el público objetivo	Sí
	5.2. Los términos utilizados son coherentes con el curso/clase	Sí
	5.3. El contenido es claro, conciso, consistente y comprensible	Sí
	5.4. Los elementos multimedia facilitan la comprensión	Sí
	5.5. Los elementos de terceros son referenciados	Sí
	5.6. Los elementos son de buena calidad	Sí
	5.7. Los elementos multimedia están integrados en los apartados y no de forma independiente	Sí
	5.8. La organización del contenido está pensada para evitar la sobrecarga cognitiva	Sí
	5.9. El contenido está disponible con facilidad	Sí
	5.10. El contenido está separado en secciones	Sí
	5.11. Es posible acceder al contenido desde múltiples vías	Sí
	5.12. El software provee de enlaces a recursos externos adicionales	Sí
	5.13. El contenido multimedia agrega valor al desarrollo del curso	Sí
6.- Experiencia de usuario	6.1. El software contiene motivación intrínseca	Sí
	6.2. El software motiva al usuario cuando realiza acciones positivas	Sí
	6.3. La retroalimentación siempre es positiva, incluso ante errores	Sí
	6.4. La retroalimentación permite la reflexión de las acciones	Sí
	6.5. El software ofrece retroalimentación según el progreso del estudiante	Sí

Nota. Elaboración propia con base en Coomans & Lacerda (2015) y Opel & Rhodes (2018).

Tras la evaluación de los 58 ítems del instrumento, fue posible identificar que la plataforma desarrollada cumple con 53 de estos (91.37% de cumplimiento). Los cinco elementos con los que no se cumplían.

Con base en la retroalimentación obtenida de las entrevistas y de la aplicación del instrumento PETESE, se agregaron dos elementos dentro de la metodología propuesta. El primero, referente al aspecto de la evaluación dentro del proceso de enseñanza-aprendizaje, mismo que, aunque fue reconocido de forma tácita en los resultados de las entrevistas, dada su importancia es indispensable incluirlo y mencionarlo de forma explícita (Llema & Vilela-Malabanan, 2019). El segundo, por su parte, hace referencia a la seguridad de la información, un concepto necesario cuando se habla de tecnologías de la información y, por supuesto, de tecnología educativa, debido al contexto en el que usualmente se utiliza el software educativo (O'Dell, 2020). Así pues, en la Figura 7.4, se presenta la metodología tras la retroalimentación.

Figura 7.4.
Metodología PADSE (post-retroalimentación)



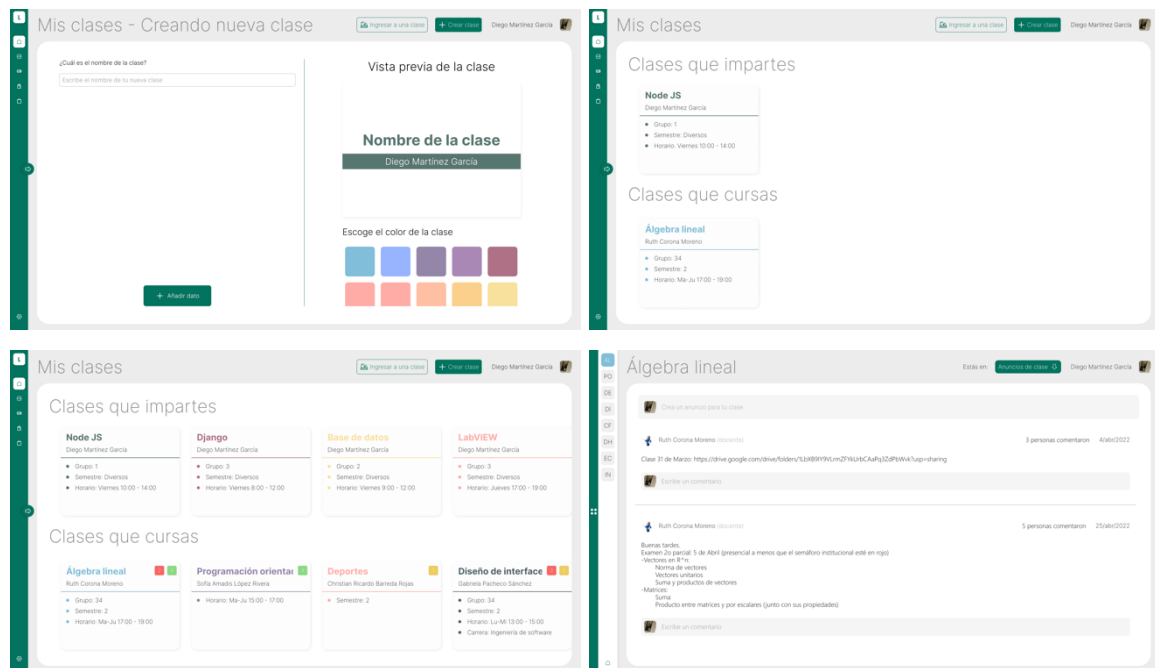
Nota. Elaboración propia con base en Adnan & Ritzhaupt (2018), Chimalakonda & Nori (2018), Coomans & Lacerda (2015), Fourcade et al. (2017), Llema & Vilela-Malabanan (2019), O'Dell (2020), Reyes (2007), Ritzhaupt & Kumar (2015), Rodríguez (2014) y Seoane & García (2014).

Es posible apreciar cómo, además de las dos modificaciones antes mencionadas, fueron identificados y agregados los actores involucrados en las diferentes etapas de la metodología. Como actores, únicamente fueron considerados los docentes y desarrolladores, en virtud de que la metodología busca que un docente, sin importar su trasfondo, tenga la capacidad de diseñar su propia plataforma de tecnología educativa sin que necesariamente se trate de un especialista en diseño instruccional o desarrollo tecnológico (Kirschner, 2015).

Esta metodología, fue puesta nuevamente a prueba mediante su aplicación en el proceso de desarrollo de una plataforma. Por lo anterior, se realizó nuevamente el diseño de una plataforma de tecnología educativa, para atender las carencias identificadas por el instrumento PETESE. En la Figura 7.5, se presenta el diseño de la segunda versión de la plataforma desarrollada.

Figura 7.5.

Segunda versión de la plataforma desarrollada



Las interfaces presentadas, dan solución a los cinco ítems que en los que se había obtenido un resultado negativo en el instrumento PETESE.

8. Discusión

La metodología PADSE, resultado de la presente investigación, no se trata de la primera ni de la única aproximación al diseño de una metodología específica para el desarrollo de software educativo. Sin embargo, sí se trata de la primera en ser diseñada mediante la integración de metodologías ágiles, principios de diseño instruccional y, como aspecto más importante, los patrones de arquitectura y de diseño de software.

Otras metodologías similares fueron desarrolladas bajo contextos en los que la tecnología aún no tenía una presencia tan ubicua como la que tiene en la actualidad, por ejemplo, la propuesta por Cataldi et al. (2003) y las mencionadas por Reyes (2007). Debido a esto, no se tienen en consideración múltiples aspectos que hoy en día son necesarios considerar y que la metodología propuesta sí contempla, como lo son canales de comunicación, experiencia de usuario, usabilidad y ecosistema tecnológico. De igual forma y a diferencia de otras aproximaciones, la incorporación de metodologías ágiles implica que el usuario final de la metodología (sea un docente o un desarrollador) tendrá una curva de aprendizaje menos pronunciada si se le compara con metodologías diseñadas o utilizadas en otros trabajos.

También, fue posible identificar un resultado inesperado. Este se encontró en las respuestas por parte de los docentes entrevistados sobre la metodología, ya que fue mencionada la posibilidad de utilizar la metodología como una herramienta para el apoyo en la selección de una o un conjunto de herramientas y plataformas ya existentes, sin necesariamente tener que desarrollar algo completamente nuevo.

El resultado inesperado puede deberse a múltiples factores, no obstante, es probable que haya surgido debido a que parte del proceso a seguir en la metodología, implica hacer una investigación, análisis, definición y planificación de las necesidades tanto pedagógicas como tecnológicas de la plataforma a desarrollar, por lo que es natural que durante este proceso se identifiquen otras herramientas que satisfagan alguna o algunas de las necesidades que plantean.

De igual forma, es necesario mencionar las limitantes de la investigación y, por supuesto, de la metodología diseñada. Primero que nada, es indispensable establecer que uno de los objetivos tácitos de la investigación, proponía que la metodología por diseñar tuviera la facilidad de ser utilizadas por docentes para involucrarse de forma directa en el proceso de desarrollo de un software educativo. Lo anterior, limita el alcance de la metodología en cuanto a su diseño, ya que no era posible utilizar conceptos especializados de alguna disciplina en particular y en específico de la Ingeniería de Software. Ya que, si bien es necesario que un docente cuente con competencias pedagógicas y tecnológicas, no es un requisito que sea un experto en tecnología. Sin embargo, por otra parte, sí es necesario que estén involucrados de forma activa en el proceso de diseño y desarrollo del software educativo (Kirschner, 2015).

Otra limitación con la que cuenta la metodología es que quien la vaya a utilizar, debe contar con competencias tecnológicas, mismas que, según el dominio que se tenga, habilitarán o inhabilitarán la posibilidad de utilizar la metodología como herramienta auxiliar en la selección de plataformas y herramientas ya desarrolladas. Aunque esto último se trate de una problemática y limitante general cuando se trata del uso de software educativo (Ertmer, 2005).

El desarrollo de esta investigación no estuvo exento de retos ya que, en la actualidad, es posible identificar diferentes requerimientos, contextos y situaciones a las que la tecnología educativa debe enfrentarse. En México, aunque la mayoría de las instituciones educativas de nivel superior contemplan la inclusión y adopción de tecnología educativa en su planeación estratégica, es un hecho que aún no existe ni la infraestructura ni los recursos suficientes para que esto se realice de forma oportuna (en especial cuando se trata de instituciones educativas públicas) (Ponce-López, Vicario-Solórzano, et al., 2022).

De igual forma, es necesario hacer la distinción entre los retos de las tecnologías de la información y los de la tecnología educativa en el país. Por una parte, los primeros implican tratar asuntos relacionados con la gestión tecnológica,

como lo son: ubicuidad, infraestructura, financiamiento, gobernanza de las tecnologías de la información y la integración de tecnologías convergentes; mientras que, los segundos, se aproxima a temas de carácter humano y social, como lo son: adopción tecnológica, competencias digitales, accesibilidad, competencias pedagógicas y capacitación (Ponce-López, Castañeda de León, et al., 2022).

Lo anterior se ve directamente reflejado en el planteamiento inicial de la metodología, misma que busca tener una curva de aprendizaje poco pronunciada y flexible, para que su aplicación sea adaptable y aplicable a diferentes contextos, en los que incluso no existan planeaciones formales o apoyos para la innovación en la docencia, además de que presenta un acercamiento integral de las necesidades tecnológicas y pedagógicas.

Finalmente, la metodología busca innovar mediante la integración de conceptos especializados de la Ingeniería de Software (patrones arquitectónicos y de diseño de software). Esta integración se realizó de forma transparente para el usuario final de la metodología, ya que, aunque no se encuentran de forma explícita, sí brindan de soporte y estructura a la metodología. Así pues, a partir de esta investigación, se presenta un nuevo paradigma desde el cual se pueden realizar aproximaciones para el desarrollo de soluciones (herramientas y plataformas) de tecnología educativa.

9. Conclusión

Desde el punto de vista teórico, el desarrollo de software educativo se trata de un tema inherentemente complejo, ya que conjuga dos disciplinas diametralmente distintas como lo son la Ingeniería de Software y el Diseño Instruccional (Ertmer, 2005). En principio, podría parecer que ambas cuentan con más similitudes que diferencias, en especial cuando se comparan sus procesos desde sus respectivas etapas (Adnan & Ritzhaupt, 2018). No obstante, es ahí donde las similitudes terminan y donde, aunque pareciera que utilizan procesos similares, la aplicación e instrumentos utilizados en cada disciplina son completamente distintos.

Algo similar sucede cuando se comparan los procesos de estas disciplinas con otros como el proceso de la Investigación Basada en el Diseño, misma que sigue una serie de etapas similar en cuanto a su estructura, aunque totalmente distintas en su aplicación (Benito & Salinas, 2016). Esto, sin embargo, no es una simple coincidencia, sino que es el resultado de la adaptación del proceso administrativo a procesos creativos para la resolución de problemas, mismos que implican el diseño y desarrollo de un producto o, en el caso de los procesos de enseñanza-aprendizaje, educar (Jiménez-Saavedra, 2014).

Ahora bien, aún con estas diferencias, fue posible reunir los principios de ambas disciplinas en un único modelo que, a su vez, sirvió de base para la metodología que se propone en el presente trabajo de investigación. Esta metodología, a través de un punto de coincidencia en los patrones de diseño y de arquitectura de software con los patrones de diseño instruccional, permitió alinear la aplicación de procesos de Ingeniería de Software con las necesidades pedagógicas inherentes a los softwares educativos (herramientas y plataformas de tecnología educativa).

A continuación, se presenta el análisis de la hipótesis, así como las respuestas a las preguntas de investigación planteadas, mismas que sirvieron como guía del proyecto de investigación.

La hipótesis planteaba que la aplicación de una metodología de desarrollo de software específica para la creación de una plataforma de tecnología educativa facilitará la implementación de modelos de diseño instruccional en dichas plataformas. Tras la realización y posterior síntesis cualitativa de las entrevistas realizadas a docentes, así como el análisis cuantitativo de la aplicación de la metodología en el diseño de una plataforma de tecnología educativa, resultó evidente que los docentes reaccionaron de forma positiva a la metodología PADSE, ya que les permite identificar con precisión qué elementos de la tecnología educativa darían soporte a lo que establecen en su proceso de diseño instruccional. En este sentido, la hipótesis resultó ser verdadera, con la limitación de que esta aún debe ser probada en contextos fuera del nivel de educación superior, ya que todos los entrevistados pertenecen a ese nivel educativo.

Ahora bien, respecto a las tres preguntas de investigación, la primera era ¿qué efecto tiene la ausencia de un sustento pedagógico en las plataformas de tecnología educativa en el proceso de enseñanza - aprendizaje? La respuesta a esta pregunta se consiguió desde la concepción de la problemática de la investigación. Se logró establecer que, como consecuencia de esta falta sustento pedagógico, los docentes tenían una percepción de insuficiencia por parte de la plataforma o herramienta de tecnología educativa que utilizaban, además de tener la necesidad de adaptarse al software utilizado, cosa que según los principios fundamentales de la Ingeniería de Software, nunca debería suceder (Opel & Rhodes, 2018).

La segunda pregunta, ¿cuál es el estado del conocimiento referente al desarrollo de plataformas de tecnología educativa? Se logró responder mediante la metasíntesis de la literatura especializada relacionada al tema de investigación, donde se encontró que los principios de la Ingeniería de Software son valorados como elementos esenciales durante el proceso de desarrollo de tecnología educativa, aunque rara vez son incluidos en las etapas de diseño, lo cual tiene un

impacto negativo sobre el resultado final, como que el producto de software sea incapaz de adaptarse a las necesidades de sus usuarios, por ejemplo.

También se planteó la siguiente interrogante, ¿qué características toman en cuenta los docentes para seleccionar la plataforma de tecnología educativa que han de utilizar para impartir sus clases? Las características que se toman en consideración fueron identificadas mediante una clasificación categórica de los conceptos relacionados con el diseño instruccional. Fue posible identificar que las características tomadas en cuenta son de corte estructural, pues se tratan de conceptos ligados a la arquitectura de un proceso de enseñanza-aprendizaje (Belloch, 2017) y, por lo tanto, hacen referencia a las capacidades que el software educativo puede ofrecer. Estas capacidades son: planeación, personalización, integración tecnológica, manejo de escenarios, mediación, motivación, evaluación y reflexión.

Finalmente, se concluye que, pese a las diferencias entre las disciplinas que conforman el producto del trabajo de investigación, no solo fue posible utilizar conceptos de disciplinas con campos semánticos diferentes, sino que también fue posible utilizarlas de forma complementaria y sinérgica. Además, una vez que la metodología fue puesta a prueba, se encontró que esta puede funcionar incluso como referencia para seleccionar herramientas y plataformas de desarrollo existentes, sin la necesidad de iniciar un proceso de desarrollo nuevo. Es así como, a partir de la metodología propuesta, se plantea la posibilidad de generar un instrumento capaz de sugerir qué plataforma, herramienta o conjunto de soluciones de tecnología educativa pueden resultar más adecuadas en un determinado escenario para un docente.

Por último, hay que mencionar que, tanto la metodología PADSE como el modelo propuesto en este trabajo de investigación, cuentan con la licencia de *Creative Commons* CC BY-NC 4.0, misma que permite que sean compartidos, adaptados y modificados mientras sean utilizados sin fines de lucro y con crédito al autor original de los mismos.

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>
- Aguña, A. G. (2012). Patrones en aprendizaje: Concepto, aplicación y diseño de un patrón. *Revista de Educación a Distancia*, 31.
- Ahmad, R., & Nisa, M. U. (2016). The significance of educational technology in teaching learning process. *The International Journal of Indian Psychology*, 4(1), 164–170.
- Angelov, S., & de Beer, P. (2017). Designing and applying an approach to software architecting in agile projects in education. *Journal of Systems and Software*, 127, 78–90. <https://doi.org/10.1016/J.JSS.2017.01.029>
- Antonenko, P. D., Dawson, K., & Sahay, S. (2017). A framework for aligning needs, abilities and affordances to inform design and practice of educational technologies. *British Journal of Educational Technology*, 48(4), 916–927. <https://doi.org/10.1111/bjet.12466>
- Area, M. (2009). Introducción a la tecnología educativa. *Universidad de la Laguna. La Laguna, España*.
- 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>
- Ausburn, L. J., & Brown, D. (2006). Learning strategy patterns and instructional preferences of career and technical education students. *Journal of STEM Teacher Education*, 43(4), 4.
- Bartolomé, A., Castañeda, L., & Adell, J. (2018). Personalisation in educational technology: the absence of underlying pedagogies. *International Journal of*

Educational Technology in Higher Education, 15(1), 14.

- Bass, L., Clements, P., & Kazman, R. (2013). *Software Architecture in Practice* (3rd ed.). Addison-Wesley Professional.
- Bayham, J., & Fenichel, E. P. (2020). Impact of school closures for COVID-19 on the US health-care workforce and net mortality: a modelling study. *The Lancet. Public health*, 5(5), e271–e278. [https://doi.org/10.1016/S2468-2667\(20\)30082-7](https://doi.org/10.1016/S2468-2667(20)30082-7)
- Belloch, C. (2017). *Diseño instruccional* (p. 15). Universidad de Valencia.
- Benito, B., & Salinas, J. (2016). La investigación basada en diseño en tecnología educativa. En *Revista Interuniversitaria de Investigación en Tecnología Educativa*. <https://doi.org/10.6018/riite2016/260631>
- Bennett, S., Agostinho, S., & Lockyer, L. (2017). The process of designing for learning: understanding university teachers' design work. *Educational Technology Research and Development*, 65(1), 125–145. <https://doi.org/10.1007/s11423-016-9469-y>
- Bennett, S., Dawson, P., Bearman, M., Molloy, E., & Boud, D. (2017). How technology shapes assessment design: Findings from a study of university teachers. *British Journal of Educational Technology*, 48(2), 672–682. <https://doi.org/10.1111/bjet.12439>
- Berggren, A., Burgos, D., Fontana, J. M., Hinkelman, D., Hung, V., Hursh, A., & Tielemans, G. (2005). Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. *Journal of Interactive Media in Education*, 2005(1), 3. <https://doi.org/10.5334/2005-2>
- Bjuhr, O., Segeljakt, K., Addibpour, M., Heiser, F., & Lagerstrom, R. (2017). Software Architecture Decoupling at Ericsson. *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings*, 259–262. <https://doi.org/10.1109/ICSAW.2017.7>

- Budoya, C. M., Kissaka, M. M., & Mtebe, J. S. (2019). Instructional Design Enabled Agile Method Using ADDIE Model and Feature Driven Development Process. *International Journal of Education and Development using Information and Communication Technology*, 20. <https://eric.ed.gov/?q=Software+design+educational+technology&id=EJ1214264>
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (2001). *Pattern-Oriented Software Architecture* (6th ed.). John Wiley & Sons, Ltd.
- Cabero, J. (2015). Reflexiones educativas sobre las tecnologías de la información y la comunicación (TIC). *Tecnología, Ciencia y Educación*, 1, 19-27.
- Camarda, P., & Minzi, V. (2012). *Primaria Digital, Aulas digitales móviles, Manual general introductorio*. Buenos Aires. Ministerio de Educación de la Nación.
- Campos, Y., Domínguez, N., & Guzmán, D. (2019). El Diseño Instruccional, una mirada desde la experiencia de la UGB ¿Cómo lo implementamos? *Proceedings of the Digital World Learning Conference CIEV*, 114–117.
- Cataldi, Z., Lage, F., Pessacq, R., & García-Martínez, R. (2003). Metodología extendida para la creación de software educativo desde una visión integradora. *Revista Latinoamericana De Tecnología Educativa*, 2(1), 10–40.
- Chen, Y., Cook, W. D., Du, J., Hu, H., & Zhu, J. (2017). Bounded and discrete data and Likert scales in data envelopment analysis: application to regional energy efficiency in China. *Annals of Operations Research*, 255(1–2), 347–366. <https://doi.org/10.1007/s10479-015-1827-3>
- Chhabra, M. (2021). A critical analysis of: Qualitative methodologies and data collection methods: Toward increased rigour in management research. *Technological Forecasting and Social Change*, 171, 120956. <https://doi.org/10.1016/J.TECHFORE.2021.120956>
- Chimalakonda, S., & Nori, K. (2018). *A Patterns Based Approach for Design of Educational Technologies*.

- Chitra, K. (2018). Selection of Design Pattern in each stage of Software Development. *Shanlax International Journal of Arts, Science and Humanities*, 5(1), 23–30.
- Clemmensen, T. (2004). Four approaches to user modelling—a qualitative research interview study of HCI professionals’ practice. *Interacting with Computers*, 16(4), 799–829. <https://doi.org/10.1016/J.INTCOM.2004.04.009>
- 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>
- Delgado, E. (2008). Metodologías de desarrollo de software. ¿Cuál es el camino? *Revista de Arquitectura e Ingeniería*, 2(3).
- Domínguez, L. C., Carreño, O., & Sierra, D. (2020). Divide y vencerás: Efectos de dos intervenciones para el aprendizaje interactivo en grupos grandes de estudiantes sobre la percepción de la calidad del ambiente de aprendizaje. *Educacion Medica*. <https://doi.org/10.1016/j.edumed.2020.04.006>
- 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>
- Englund, C., Olofsson, A. D., & Price, L. (2017). Teaching with technology in higher education: understanding conceptual change and development in practice. *Higher Education Research and Development*, 36(1), 73–87. <https://doi.org/10.1080/07294360.2016.1171300>
- Ertmer, P. A. (2005). Teacher pedagogical beliefs: The final frontier in our quest for technology integration? *Educational Technology Research and Development*, 53(4), 25–39. <https://doi.org/10.1007/BF02504683>
- Exter, M. (2018). Developing Multi-Disciplinary Skills through a Course in Educational Software Design. *International Journal of Designs for Learning*, 9(1), 49–79. <https://doi.org/10.14434/ijdl.v9i1.23413>

- Ferrer, M. L., & Madriz, F. S. (2009). Enfoques teóricos y definiciones de la tecnología educativa en el siglo XX. *Revista Electrónica" Actualidades Investigativas en Educación"*, 9(2), 1–29.
- FitzGerald, E., Kucirkova, N., Jones, A., Cross, S., Ferguson, R., Herodotou, C., Hillaire, G., & Scanlon, E. (2018). Dimensions of personalisation in technology-enhanced learning: A framework and implications for design. *British Journal of Educational Technology*, 49(1), 165–181. <https://doi.org/10.1111/bjet.12534>
- Fourcade, L. R., Arakaki, L., Riesco, D. E., Montejano, G. A., & Debnath, N. C. (2017). Extendiendo transformaciones MDA con metamodelo de patrones de diseño. *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software* (1ra ed.). Addison-Wesley Professional.
- García, G. (2020). *Temas de introducción a la formación pedagógica* (1ra ed.). Editorial Pueblo y Educación.
- Gómez Díaz de León, C. (2014). Estudio comparado. En *Métodos y técnicas cualitativas y cuantitativas aplicables a las investigaciones en ciencias sociales* (pp. 224–251). Tirant Humanidades.
- Gros, B. (2012). Retos y tendencias sobre el futuro de la investigación acerca del aprendizaje con tecnologías digitales. *Revista de Educación a Distancia*, 32.
- Gros, B., Escofet, A., & Marimón, M. (2016). Los patrones de diseño como herramientas para guiar la práctica del profesorado. *Revista Latinoamericana de Tecnología Educativa-RELATEC*, 15(3), 11–25.
- Groth, R. E. (2017). Developing Statistical Knowledge for Teaching during Design-Based Research. *Statistics Education Research Journal*, 16(2).
- Heeager, L. T., & Nielsen, P. A. (2018). A conceptual model of agile software

- development in a safety-critical context: A systematic literature review. *Information and Software Technology*, 103, 22–39. <https://doi.org/10.1016/J.INFSOF.2018.06.004>
- Henderson, M., Selwyn, N., & Aston, R. (2017). What works and why? Student perceptions of ‘useful’ digital technology in university teaching and learning. *Studies in Higher Education*, 42(8), 1567–1579. <https://doi.org/10.1080/03075079.2015.1007946>
- Hew, K. F., Lan, M., Tang, Y., Jia, C., & Lo, C. K. (2019). Where is the “theory” within the field of educational technology research? *British Journal of Educational Technology*, 50(3), 956–971. <https://doi.org/10.1111/bjet.12770>
- Holt, L. C., & Kysilka, M. (2005). *Instructional patterns: Strategies for maximizing student learning*. Sage Publications.
- Hoogveld, A. W. M., Paas, F., Jochems, W. M. G., & Van Merriënboer, J. J. G. (2002). Exploring teachers’ instructional design practices from a systems design perspective. *Instructional Science*, 30(4), 291–305. <https://doi.org/10.1023/A:1016081812908>
- Hughes, K. L., Williamson, P. R., & Young, B. (2022). In-depth qualitative interviews identified barriers and facilitators that influenced chief investigators’ use of core outcome sets in randomised controlled trials. *Journal of Clinical Epidemiology*, 144, 111–120. <https://doi.org/10.1016/J.JCLINEPI.2021.12.004>
- Hussain, S., Keung, J., & Khan, A. A. (2017). Software design patterns classification and selection using text categorization approach. *Applied Soft Computing Journal*, 58, 225–244. <https://doi.org/10.1016/j.asoc.2017.04.043>
- 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*, 62–75. <https://doi.org/10.6018/RIITE.463421>
- Ipek, I., & Ziatdinov, R. (2018). *New Approaches and Trends in the Philosophy of*

Educational Technology for Learning and Teaching Environments.
<https://doi.org/10.13187/ejced.2017.3.381>

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>

Jiménez-Saavedra, S. (2014). Tecnología educativa: campos de formación y perfil diferencial. *Revista Iberoamericana de Educación Superior*, 5(14), 125a – 141.
[https://doi.org/10.1016/s2007-2872\(14\)70304-x](https://doi.org/10.1016/s2007-2872(14)70304-x)

Kahle, D. (2008). Designing open educational technology. *Opening up education: The collective advancement of education through open technology, open content, and open knowledge*, 27–45.

Khomh, F., & Gueheneuc, Y. G. (2018). Design patterns impact on software quality: Where are the theories? *25th IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2018 - Proceedings, 2018-March*, 15–25. <https://doi.org/10.1109/SANER.2018.8330193>

Kirschner, P. A. (2015). Do we need teachers as designers of technology enhanced learning? *Instructional Science*, 43(2), 309–322.
<https://doi.org/10.1007/s11251-015-9346-9>

Koehler, M. J., & Mishra, P. (2005). What Happens When Teachers Design Educational Technology? The Development of Technological Pedagogical Content Knowledge. *Journal of Educational Computing Research*, 32(2), 131–152. <https://doi.org/10.2190/0EW7-01WB-BKHL-QDYV>

Koehler, N. A., Thompson, A. D., Correia, A. P., & Hagedorn, L. S. (2015). Designing online software for teaching the concept of variable that facilitates mental interaction with the material: systemic approach. *Educational Technology Research and Development*, 63(1), 97–124. <https://doi.org/10.1007/s11423-014-9357-2>

- 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/cmj.2016.1>
- Laurillard, D., Kennedy, E., Charlton, P., Wild, J., & Dimakopoulos, D. (2018). Using technology to develop teachers as designers of TEL: Evaluating the learning designer. *British Journal of Educational Technology*, 49(6), 1044–1058. <https://doi.org/10.1111/bjet.12697>
- 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>
- Lee, J., Lim, C., & Kim, H. (2017). Development of an instructional design model for flipped learning in higher education. *Educational Technology Research and Development*, 65(2), 427–453. <https://doi.org/10.1007/s11423-016-9502-1>
- Llema, C. F., & Vilela-Malabanan, C. M. (2019). Design and development of MLERWS: A user-centered mobile application for English reading and writing skills. *Procedia Computer Science*, 161, 1002–1010. <https://doi.org/10.1016/j.procs.2019.11.210>
- López, V. (2018). *Método de Diseño Instruccional para Generar Entornos Virtuales de Aprendizaje en Línea* [Universidad Autónoma de Querétaro]. <http://ri-ng.uaq.mx/handle/123456789/1044>
- Lorenzo-Lledó, A. (2018). Innovación en el aprendizaje desde el diseño tecnopedagógico. *International Studies on Law and Education*, 29–30, 119–130. http://www.hottopos.com/isle29_30/119-130Lorenzo.pdf
- Lowyck, J. (2014). Bridging learning theories and technology-enhanced environments: A critical appraisal of its history. En *Handbook of Research on Educational Communications and Technology: Fourth Edition* (pp. 3–20).

Springer New York. https://doi.org/10.1007/978-1-4614-3185-5_1

- Luna-Gijón, G., & Porras-Hernández, L. H. (2014). Validación de un modelo instruccional centrado en el diseño de materiales digitales de aprendizaje. *Investigación en Educación Médica*, 3(11), 123–130. [https://doi.org/10.1016/s2007-5057\(14\)72738-3](https://doi.org/10.1016/s2007-5057(14)72738-3)
- Marín, V. I., Duarte, J. M., Galvis, A. H., & Zawacki-Richter, O. (2018). Thematic analysis of the international journal of educational Technology in Higher Education (ETHE) between 2004 and 2017. En *International Journal of Educational Technology in Higher Education* (Vol. 15, Número 1, pp. 1–7). Springer Netherlands. <https://doi.org/10.1186/s41239-018-0089-y>
- Marín, V. I., Zawacki-Richter, O., Pérez, A., & Salinas, J. (2017). Educational Technology trends in the Ibero-American world: 20 years of the Edutec-e journal. *Edutec. Revista Electrónica de Tecnología Educativa*, 59. <https://doi.org/10.21556/edutec.2017.59.836>
- Martinez-Maldonado, R., Goodyear, P., Carvalho, L., Thompson, K., Hernandez-Leo, D., Dimitriadis, Y., Prieto, L. P., & Wardak, D. (2017). Supporting collaborative design activity in a multi-user digital design ecology. *Computers in Human Behavior*, 71, 327–342. <https://doi.org/10.1016/j.chb.2017.01.055>
- McKenney, S., Kali, Y., Markauskaite, L., & Voogt, J. (2015). Teacher design knowledge for technology enhanced learning: an ecological framework for investigating assets and needs. *Instructional Science*, 43(2), 181–202. <https://doi.org/10.1007/s11251-014-9337-2>
- McKnight, K., O'Malley, K., Ruzic, R., Horsley, M., Franey, J. J., & Bassett, K. (2016). Teaching in a digital age: How educators use technology to improve student learning. *Journal of Research on Technology in Education*, 48(3), 194–211. <https://doi.org/10.1080/15391523.2016.1175856>
- Meneses, G. (2007). El proceso de enseñanza–aprendizaje: el acto didáctico. En *NTIC, Interacción y aprendizaje en la universidad*. (pp. 31–65).

- Mishra, S. (2002). A design framework for online learning environments. *British Journal of Educational Technology*, 33(4), 493–496. <https://doi.org/10.1111/1467-8535.00285>
- Mohammadi, H. (2015). Investigating users' perspectives on e-learning: An integration of TAM and IS success model. *Computers in Human Behavior*, 45, 359–374. <https://doi.org/10.1016/j.chb.2014.07.044>
- Navarro-Cadavid, A., Fernández-Martínez, J. D., & Morales-Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11(2), 30–39.
- Navarro, M. E., Moreno, M. P., Aranda, J., Parra, L., Rueda, J. R., & Pantano, J. C. (2017). Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles. *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*.
- Nguyen, G. N. H., & Bower, M. (2018). Novice teacher technology-enhanced learning design practices: The case of the silent pedagogy. *British Journal of Educational Technology*, 49(6), 1027–1043. <https://doi.org/10.1111/bjet.12681>
- Niederhauser, D. S., Howard, S. K., Voogt, J., Agyei, D. D., Laferriere, T., Tondeur, J., & Cox, M. J. (2018). Sustainability and Scalability in Educational Technology Initiatives: Research-Informed Practice. *Technology, Knowledge and Learning*, 23(3), 507–523. <https://doi.org/10.1007/s10758-018-9382-z>
- 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>
- O'Dell, K. (2020). Modern Marginalia: Using Digital Annotation in the Composition Classroom. *Computers and Composition*, 56, 102570. <https://doi.org/10.1016/j.compcom.2020.102570>
- Opel, D. S., & Rhodes, J. (2018). Beyond Student as User: Rhetoric, Multimodality,

- and User-Centered Design. *Computers and Composition*, 49, 71–81.
<https://doi.org/10.1016/J.COMPCOM.2018.05.008>
- Oropeza, D. K., Rengifo, Y. M., & Parra, D. (2016). Tecnología educativa para la tutoría-orientación universitaria en estudios virtuales. *Novum Scientiarum*, 4.
- Ortega Barba, C. F. (2014). Information technology and communication for educational innovation. En *Perfiles Educativos* (Vol. 36, Número 144, pp. 214–218). Universidad Nacional Autónoma de México.
[https://doi.org/10.1016/S0185-2698\(14\)70633-6](https://doi.org/10.1016/S0185-2698(14)70633-6)
- 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>
- Özcan-Top, Ö., & Demirors, O. (2019). Application of a software agility assessment model – AgilityMod in the field. *Computer Standards & Interfaces*, 62, 1–16.
<https://doi.org/10.1016/J.CSI.2018.07.002>
- Palacios Vicario, B., Sánchez Gómez, M. C., & Gutiérrez García, A. (2013). Evaluar la calidad en la investigación cualitativa: Guías o checklists. *2º Congreso Nacional sobre Metodología de la Investigación en Comunicación*, 581–596.
- Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. *Procedia - Social and Behavioral Sciences*, 175, 455–463.
<https://doi.org/10.1016/J.SBSPRO.2015.01.1223>
- Parmaxi, A., Zaphiris, P., & Ioannou, A. (2016). Enacting artifact-based activities for social technologies in language learning using a design-based research approach. *Computers in Human Behavior*, 63, 556–567.
<https://doi.org/10.1016/j.chb.2016.05.072>
- Piza Burgos, N. D., Amaiquema Márquez, F. A., Beltrán Baquerizo, G. E., Piza Burgos, N. D., Amaiquema Márquez, F. A., & Beltrán Baquerizo, G. E. (2019).

- Métodos y técnicas en la investigación cualitativa. Algunas precisiones necesarias. *Conrado*, 15(70), 455–459.
- Ponce-López, J. L., Castañeda de León, L. M., & López-Valencia, F. (2022). *Estado actual de las tecnologías de la información y comunicación en las instituciones de educación superior en México*. Asociación Nacional de Universidades e Instituciones de Educación Superior.
- Ponce-López, J. L., Vicario-Solórzano, C. M., & López-Valencia, F. (2022). *Estado actual de las tecnologías educativas en las instituciones de educación superior en México* (1st ed.). Asociación Nacional de Universidades e Instituciones de Educación Superior.
- Potkonjak, V., Gardner, M., Callaghan, V., Mattila, P., Guetl, C., Petrović, V. M., & Jovanović, K. (2016). Virtual laboratories for education in science, technology, and engineering: A review. *Computers and Education*, 95, 309–327. <https://doi.org/10.1016/j.compedu.2016.02.002>
- Pressman, R. S., & Troya, J. M. (2010). *Ingeniería del software* (7a ed.). McGraw Hill.
- Reyes, M. (2007). *Metodologías para el desarrollo de software educativo*. Universidad Autónoma del Estado de Hidalgo.
- Riaz, M. N. (2018). Impact of software design patterns on the quality of software: A comparative study. *2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings, 2018-Janua*, 1–6. <https://doi.org/10.1109/ICOMET.2018.8346325>
- Richards, M. (2015). *Software Architecture Patterns* (H. Scherer & C. Lobner (eds.); 3rd ed.). O'Reilly Media.
- 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>

- Robinson, R., Molenda, M., & Rezabek, L. (2013). Facilitating learning. En *Educational Technology* (pp. 27–60). Routledge.
- Rodríguez, N. (2014). Fundamentos Del Proceso Educativo a Distancia: Enseñanza, Aprendizaje Y Evaluación. *RIED. Revista Iberoamericana de Educación a Distancia*, 17(2), 75–93. <https://doi.org/10.5944/ried.17.2.12679>
- 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>
- San Martín, Á., & García, Á. (2016). Pedagogic dilemmas to flows of knowledge in the age of digital technology. *International Journal of Educational Technology in Higher Education*, 13(1), 1–9. <https://doi.org/10.1186/s41239-016-0030-1>
- Sánchez-Meca, J. (2010). Cómo realizar una revisión sistemática y un meta-análisis. *Aula Abierta*, 38(2), 53–64.
- Sánchez, M. T. R., Ramiro-Sánchez, T., & Alba-Ruiz, R. (2014). Perfil bibliométrico de RUSC. *Universities and Knowledge Society Journal. RUSC Universities and Knowledge Society Journal*, 11(3), 44–66. <https://doi.org/10.7238/rusc.v11i3.2167>
- Seckler, M., Heinz, S., Forde, S., Tuch, A. N., & Opwis, K. (2015). Trust and distrust on the web: User experiences and website characteristics. *Computers in Human Behavior*, 45, 39–50. <https://doi.org/10.1016/J.CHB.2014.11.064>
- Seoane, A. M. P., & García, F. J. G. (2014). Patrones pedagógicos y docencia en red. *Formación en Red: Aprender con tecnologías digitales*, 30–47.
- Serrano, J. L., Gutiérrez, I., & Prendes, M. P. (2016). Internet como recurso para enseñar y aprender. Una aproximación práctica a la tecnología educativa. *Sevilla: Eduforma*.
- Sharif, A., & Cho, S. (2015). Diseñadores instruccionales del siglo XXI: Cruzando

- las brechas perceptuales entre la identidad, práctica, impacto y desarrollo profesional. *RUSC Universities and Knowledge Society Journal*, 12(3), 72–86. <https://doi.org/10.7238/rusc.v12i3.2176>
- Spector, J. M. (2017). Reflections on educational technology research and development. En *Educational Technology Research and Development* (Vol. 65, Número 6, pp. 1415–1423). Springer New York LLC. <https://doi.org/10.1007/s11423-017-9545-y>
- Strecker, S., Kundisch, D., Lehner, F., Leimeister, J. M., & Schubert, P. (2018). Higher Education and the Opportunities and Challenges of Educational Technology. *Business & Information Systems Engineering*, 60(2), 181–189. <https://doi.org/10.1007/s12599-018-0522-8>
- Tai, Y., & Ting, Y. L. (2020). English-learning mobile app designing for engineering students' cross-disciplinary learning and collaboration: A sample practice and preliminary evaluation. *Australasian Journal of Educational Technology*, 36(4), 120–136. <https://doi.org/10.14742/ajet.4999>
- Thelwall, M., & Nevill, T. (2021). Is research with qualitative data more prevalent and impactful now? Interviews, case studies, focus groups and ethnographies. *Library & Information Science Research*, 43(2), 101094. <https://doi.org/10.1016/J.LISR.2021.101094>
- 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>
- Tirado, O., Luis, J., Astorga, I., Alejandro, J., Mendoza, L., Carlos, J., Ramírez, R., Ulises, R., & Garzón, Á. P. (2019). Clasificación de los patrones de diseño idóneos en programación Android. En *Revista Digital de Tecnologías Informáticas y Sistemas* (Vol. 3, Número 3).
- van Vliet, H., & Tang, A. (2016). Decision making in software architecture. *Journal of Systems and Software*, 117, 638–644.

<https://doi.org/10.1016/j.jss.2016.01.017>

Viner, R. M., Russell, S. J., Croker, H., Packer, J., Ward, J., Stansfield, C., Mytton, O., Bonell, C., & Booy, R. (2020). School closure and management practices during coronavirus outbreaks including COVID-19: a rapid systematic review. En *The Lancet Child and Adolescent Health* (Vol. 4, Número 5, pp. 397–404). Elsevier B.V. [https://doi.org/10.1016/S2352-4642\(20\)30095-X](https://doi.org/10.1016/S2352-4642(20)30095-X)

Waight, N., Liu, X., & Gregorius, R. M. (2015). Understanding the life cycle of computer-based models: the role of expert contributions in design, development and implementation. *Educational Technology Research and Development*, 63(6), 831–859. <https://doi.org/10.1007/s11423-015-9402-9>

Williams, L. (2010). Agile Software Development Methodologies and Practices. *Advances in Computers*, 80, 1–44. [https://doi.org/10.1016/S0065-2458\(10\)80001-4](https://doi.org/10.1016/S0065-2458(10)80001-4)

Zawacki-Richter, O., Alturki, U., & Aldraiweesh, A. (2017). Review and content analysis of the International Review of Research in Open and Distance/Distributed Learning (2000-2015). En *International Review of Research in Open and Distance Learning* (Vol. 18, Número 2, pp. 1–26). Athabasca University. <https://doi.org/10.19173/irrodl.v18i2.2806>

Anexos

Anexo 1. Encuesta para la definición del problema de la investigación

Demográficos (D)

1. ¿Cuántos años tiene dando clases?
 - Entre 0 y 2 años
 - Entre 3 y 5 años
 - Entre 6 y 8 años
 - Más de 9 años

2. Sexo
 - Masculino
 - Femenino

3. ¿Qué edad tiene?
 - < 23 años
 - 24 – 27 años
 - 27 – 30 años
 - 30 – 33 años
 - > 34 años

4. ¿A qué área del conocimiento pertenece su licenciatura?
 - Tecnologías de la Información y Comunicación
 - Ciencias Sociales
 - Contaduría y Administración
 - Ciencias de la Salud
 - Ciencias Naturales
 - Ingeniería

Diseño Instruccional (DI)

5. ¿Con qué regularidad realiza alguna planeación para impartir sus clases? (cronograma de actividades, elaboración de recursos y materiales, metodología de enseñanza, etc.).
 - Nunca
 - Casi nunca
 - Algunas veces

- Casi siempre
 - Siempre
6. ¿Con qué frecuencia utiliza alguna plataforma de tecnología educativa para las clases que imparte?
- Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre
7. ¿Cuánto tiempo le dedica al diseño y planeación de sus clases?
- < 1 hora
 - Entre 1 y 2 horas
 - Entre 2 y 3 horas
 - Entre 3 y 4 horas
 - > 4 horas

Diseño Instruccional Digital

8. ¿Cuántas veces ha tenido que modificar la planeación de la clase debido a la plataforma de tecnología educativa que utiliza?
- Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre
9. ¿Con qué frecuencia realiza alguna planeación especial para la plataforma de tecnología educativa que emplea?
- Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre
10. ¿Cuánto tiempo le dedica al diseño y planeación de sus clases cuando emplea una plataforma de tecnología educativa?
- < 1 hora

- Entre 1 y 2 horas
- Entre 2 y 3 horas
- Entre 3 y 4 horas
- > 4 horas

Aplicación de Tecnología Educativa (ATE)

11. ¿Con qué frecuencia ha utilizado las Google Apps for Education (Classroom, Drive, Hangouts) en las clases que imparte?
 - Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre
12. ¿Con qué frecuencia ha utilizado Microsoft Education Suite (Teams, OneDrive, Office 365) en las clases que imparte?
 - Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre
13. ¿Con qué frecuencia ha utilizado el Campus Virtual de la UAQ en las clases que imparte?
 - Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre
14. ¿Con qué frecuencia ha utilizado Canvas Instructure en las clases que imparte?
 - Nunca
 - Casi nunca
 - Algunas veces
 - Casi siempre
 - Siempre

15. ¿Con qué frecuencia ha utilizado Kahoot! en las clases que imparte?

- Nunca
- Casi nunca
- Algunas veces
- Casi siempre
- Siempre

16. ¿Con qué frecuencia ha utilizado Tinkercad en las clases que imparte?

- Nunca
- Casi nunca
- Algunas veces
- Casi siempre
- Siempre

Percepción de las plataformas de tecnología educativa (PTE)

17. ¿Qué tan satisfecho se encuentra con la plataforma que utiliza actualmente?

- No satisfecho
- Poco satisfecho
- Moderadamente satisfecho
- Muy satisfecho
- Extremadamente satisfecho

18. ¿Qué tan satisfechos ha notado a sus alumnos con la aplicación de una plataforma de tecnología educativa?

- No satisfechos
- Poco satisfechos
- Moderadamente satisfechos
- Muy satisfechos
- Extremadamente satisfechos

19. ¿Qué tan de acuerdo está con la expresión: “Las plataformas de tecnología educativa cumplen con mis expectativas como docente”?

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

20. ¿Qué tan de acuerdo está con la expresión: ¿Los recursos tecnológicos permiten una mayor interacción con el conocimiento, motivando el proceso de aprendizaje?
- Totalmente en desacuerdo
 - En desacuerdo
 - Ni de acuerdo ni en desacuerdo
 - De acuerdo
 - Totalmente de acuerdo
21. Desde su punto de vista, ¿qué importancia tiene la utilización de recursos tecnológicos, como apoyo didáctico en los procesos de enseñanza y de aprendizaje?
- Nada importante
 - Importante
 - Regular
 - Poco importante
 - Muy importante

Implementación de diseño instruccional en plataformas de tecnología educativa (IDT)

22. ¿Qué tan de acuerdo está con la expresión: “Considero que las plataformas de tecnología educativa no me permiten desarrollar plenamente las clases que imparto”?
- Totalmente en desacuerdo
 - En desacuerdo
 - Ni de acuerdo ni en desacuerdo
 - De acuerdo
 - Totalmente de acuerdo
23. ¿En qué grado concuerda con la siguiente expresión: “Las plataformas de tecnología educativa que existen actualmente limitan el tipo de enseñanza que se puede impartir”?
- Totalmente en desacuerdo
 - En desacuerdo
 - Ni de acuerdo ni en desacuerdo
 - De acuerdo
 - Totalmente de acuerdo

24. ¿Qué tan de acuerdo está con la expresión: “La plataforma de tecnología educativa me provee de toda la información que necesito de mis alumnos”?
- Totalmente en desacuerdo
 - En desacuerdo
 - Ni de acuerdo ni en desacuerdo
 - De acuerdo
 - Totalmente de acuerdo

Sustento pedagógico para el diseño de tecnología educativa (SPTE)

25. ¿Qué tan de acuerdo está con la expresión: “Puedo aplicar perfectamente la planeación de mi materia en la plataforma de tecnología educativa que utilizo”?
- Totalmente en desacuerdo
 - En desacuerdo
 - Ni de acuerdo ni en desacuerdo
 - De acuerdo
 - Totalmente de acuerdo
26. ¿Qué tanto concuerda con lo siguiente: “Tengo que adecuar mi planeación del curso a lo que la plataforma de tecnología educativa me permite hacer”?
- Totalmente en desacuerdo
 - En desacuerdo
 - Ni de acuerdo ni en desacuerdo
 - De acuerdo
 - Totalmente de acuerdo
27. ¿Qué tan fácil o difícil le resulta implementar la planeación de una clase en una plataforma de tecnología educativa?
- Muy difícil
 - Difícil
 - Regular
 - Fácil
 - Muy fácil

28. Indique los factores que a su parecer son los que más complican el uso una plataforma de tecnología educativa:

- Falta de preparación
- Desconocimiento de la plataforma
- Limitaciones de la plataforma
- Dificultad de implementar una metodología de enseñanza
- Falta de concordancia entre la plataforma y la forma de enseñar
- Falta de recursos didácticos dentro de la plataforma

Diseño de software (DS)

29. ¿Cómo considera que son las opciones de tecnología educativa que emplea ofrecen medios alternativos suficientes para desarrollar un curso bajo diferentes condiciones? (por ejemplo, cuando no se dispone de una conexión a internet).

- Excelentes
- Muy buenas
- Regulares
- Malas
- Muy malas

30. ¿Qué tan de acuerdo está con la expresión: “Las plataformas de tecnología educativa que he utilizado hasta el momento parecen estar diseñadas bajo un sustento pedagógico”?

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

31. ¿Con qué frecuencia se ve limitado su diseño de clase por las capacidades de la plataforma de tecnología educativa?

- Nunca
- Casi nunca
- Algunas veces
- Casi siempre
- Siempre

Anexo 2. Entrevista para evaluación de la metodología desarrollada

Buenos días/tardes/noches. Por motivos del registro, me presento, soy Mauricio Arturo Ibarra Corona, ¿le podría realizar una entrevista?, muchas gracias. ¿Tiene inconveniente en que esta entrevista sea grabada para su posterior análisis?, gracias. Entonces, para comenzar, me podría decir su nombre, edad, ocupación, último grado académico que tiene y en qué área imparte clases.

1. ¿Le resulta sencillo planear sus clases utilizando tecnología educativa (ej. Google Classroom)? (a, b, d)
2. ¿Qué lo impulsa a utilizar la plataforma que utiliza en lugar de otra? (a, b)
3. ¿Qué es lo que más le desagrada de utilizar una plataforma de tecnología educativa? ¿Por qué? (a, b)
4. Para usted, ¿cuáles son las ventajas que tiene el uso de las plataformas de tecnología educativa? (a, g, f)
5. Desde su punto de vista profesional, ¿qué opina de los elementos y pasos que contempla? (c, e, f)
6. ¿Qué le agregaría o modificaría? (b, d, e, f)
7. ¿Cómo utilizaría la metodología en su labor como profesor(a)? (a, b, e, f)
8. ¿Qué le aporta la metodología a su quehacer docente? (a,b, e, f)

Categorías

- a) PTE - Percepción de las Plataformas de Tecnología Educativa.
- b) ATE - Aplicación de Tecnología Educativa.
- c) DID - Diseño Instruccional Digital.
- d) IDT - Implementación de Diseño Instruccional en la Tecnología Educativa.
- e) SPTE - Sustento Pedagógico.
- f) DS - Diseño de Software.

(las categorías se encuentran preasignadas a las preguntas, sin embargo, durante el análisis de las entrevistas, no se descarta la aparición de nuevas categorías o de categorías existentes en preguntas donde no se encontraban)