

## **Análisis comparativo de metodologías de desarrollo de software multimedia bajo el estándar ISO/IEC/IEEE 12207: un enfoque basado en el mapeo de procesos**

Giannina Núñez Marín<sup>(1)</sup>

ORCID iD: 0000-0003-4436-3703

Correo electrónico: [giannina.nunez@up.ac.pa](mailto:giannina.nunez@up.ac.pa)

Diego Santimateo Gálvez<sup>(1)</sup>

ORCID iD: 0000-0002-1999-1743

Correo electrónico: [diego.santimateo@up.ac.pa](mailto:diego.santimateo@up.ac.pa)

Carmen C. Cortés Campos<sup>(1)</sup>

ORCID iD: 0000-0003-2269-0797

Correo electrónico: [carmen.cortez@up.ac.pa](mailto:carmen.cortez@up.ac.pa)

Yahaira Y. Juárez Ortega<sup>(1)</sup>

ORCID iD: 0000-0002-3973-6758

Correo electrónico: [yahaira.juarez@up.ac.pa](mailto:yahaira.juarez@up.ac.pa)

<sup>(1)</sup> Facultad de Informática, Electrónica y Comunicación, Centro Regional Universitario de Veraguas – Universidad de Panamá

### **Resumen**

El desarrollo de software multimedia implica características distintivas relacionadas con la integración de diversos tipos de datos, la interactividad y la experiencia del usuario, aspectos que con frecuencia no reciben suficiente atención por parte de las metodologías tradicionales de ingeniería de software. Este estudio realizó un análisis comparativo de las fases del ciclo de vida en ocho metodologías de desarrollo de software multimedia, en relación con los procesos técnicos definidos por el estándar ISO/IEC/IEEE 12207. El análisis identificó los métodos de ingeniería de software aplicados, el dominio de contenido de aplicación, las prácticas de validación y las técnicas de documentación recomendadas. La metodología empleada se basó en la recopilación de datos provenientes de diversas fuentes bibliográficas, la selección de metodologías relevantes (1990–2025) y el mapeo de procesos para alinear las fases del ciclo de vida de cada metodología con los procesos técnicos del estándar ISO/IEC/IEEE 12207. Los resultados mostraron que las metodologías analizadas cubren entre el 50 % y el 75 % de los procesos técnicos del estándar, dejando vacíos importantes en las áreas de validación y documentación. Una de las principales limitaciones del estudio radica en la ausencia de validación empírica en proyectos reales, lo que resalta la necesidad de desarrollar estudios de caso futuros y herramientas automatizadas de mapeo. Los hallazgos ofrecen criterios científicos para la selección y adaptación de metodologías de desarrollo de software multimedia, contribuyendo a mejorar la calidad de productos de software tanto educativos como comerciales.

**Palabras clave:** Ciclo de vida del software, ISO/IEC/IEEE 12207, mapeo de procesos, metodología de desarrollo de software, software multimedia

## Introducción

El desarrollo de software multimedia implica consideraciones particulares debido a los tipos de datos que gestiona y a las características funcionales del propio software, ambos factores que influyen significativamente en la experiencia del usuario. La funcionalidad del software depende de la interacción entre los usuarios y los dispositivos periféricos, lo que implica que la ejecución de la aplicación se condiciona por las decisiones del usuario durante la interacción con distintos elementos o personajes dentro del entorno.

Los productos multimedia se clasifican en dos categorías: interactivos y no interactivos. En este contexto, el interés se centra en los productos multimedia interactivos—aplicaciones de software que integran componentes multimedia, como aplicaciones basadas en eventos, aplicaciones web multimedia, materiales de aprendizaje multimedia interactivos, sistemas de comunicación multimedia y sistemas de entretenimiento multimedia (Al-Jabari et al., 2019; Hannington & Reed, 2002).

En términos generales, el software opera como un sistema regido por un ciclo de vida; por tanto, las etapas desde su concepción hasta su retiro deben definirse dentro de metodologías compuestas por procesos y actividades de desarrollo, que a su vez actúan como marcos de referencia para la comunicación y la comprensión (ISO/IEC/IEEE, 2017). La ingeniería de software implica la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software. Esta disciplina abarca procesos, métodos y herramientas orientadas a apoyar la creación de sistemas computacionales complejos (ISO/IEC/IEEE, 2017; Pressman, 2010). El desarrollo de software, por lo general, emplea metodologías—entendidas como sistemas de prácticas, técnicas, procedimientos y reglas—junto con métodos, que consisten en procedimientos estructurados para lograr entregables a lo largo del ciclo de vida del software, y modelos, que funcionan como abstracciones destinadas a apoyar la comprensión, el diseño y la comunicación del software (Project Management Institute, 2021; Washizaki, 2024).

En el ámbito del software multimedia, diversas metodologías abordan las fases del ciclo de vida del software. Estos enfoques buscan responder a los desafíos particulares del desarrollo de aplicaciones multimedia al considerar aspectos como la interactividad, la usabilidad y la experiencia del usuario. En contextos educativos orientados al desarrollo de aplicaciones multimedia, los estudiantes suelen enfrentar vacíos relacionados con las prácticas de documentación y la gestión del ciclo de vida de este tipo de software. Como señalan Al-Jabari et al. (2019), los estudiantes experimentan dificultades al analizar, diseñar y desarrollar contenido multimedia para sus proyectos, debido principalmente a la ausencia de una metodología ampliamente adoptada para el desarrollo de productos multimedia. En consecuencia, tienden a adoptar metodologías tradicionales de ingeniería de software—como los modelos en cascada, incremental o ágil—que no abordan de manera adecuada los requerimientos específicos del desarrollo multimedia. Muchos modelos de procesos de software enmascaran las características propias del dominio mediante terminología genérica orientada a describir marcos de referencia generalizados (Hannington & Reed, 2002).

La norma ISO/IEC/IEEE 12207:2017 sobre ingeniería de sistemas y software proporciona un marco común para los procesos del ciclo de vida del software y establece una terminología estandarizada para la industria. Esta norma delimita los procesos, actividades y tareas aplicables a la adquisición, suministro, desarrollo, operación, mantenimiento y retiro de sistemas, productos y servicios de software. No obstante, la norma no prescribe un modelo de ciclo de vida, metodología

de desarrollo ni técnica específica, lo cual otorga a las organizaciones la flexibilidad necesaria para seleccionar y aplicar los enfoques que mejor se ajusten a sus necesidades particulares (ISO/IEC/IEEE, 2017).

No obstante, la norma destaca varios elementos clave que pueden vincularse con las metodologías de desarrollo de software, tales como:

- Un marco común de procesos para describir el ciclo de vida del software.
- Un enfoque estructurado que integre disciplinas especializadas y equipos dentro de un esfuerzo colaborativo.
- La definición sistemática de necesidades, la documentación de requisitos, el diseño del sistema y la validación dentro del ciclo de desarrollo.

Este estudio tuvo como objetivo realizar un análisis comparativo de diversas metodologías de desarrollo de software multimedia en función de los procesos técnicos definidos en la norma ISO/IEC/IEEE 12207. Estudios previos han abordado metodologías de desarrollo de software de propósito general a partir de diferentes criterios analíticos, tales como el enfoque de desarrollo aplicado (secuencial, incremental, evolutivo o ágil) (Mujumdar et al., 2012); las fortalezas y debilidades en relación con la estabilidad de los requisitos y la complejidad del proyecto (Despa, 2014; Gbaranwi et al., 2021; Islam & Ferworn, 2020; Saleh et al., 2017); así como estudios comparativos entre metodologías ágiles y en cascada orientados a construir árboles de decisión para guiar la selección metodológica (Mishra & Alzoubi, 2023). Tetteh (2024) analizó modelos ágiles de desarrollo de software con el propósito de identificar sus funcionalidades, fortalezas y limitaciones, ofreciendo recomendaciones para la toma de decisiones en la industria. Su estudio destacó desafíos relevantes, como la escalabilidad, la integración de DevOps, la toma de decisiones basada en datos, la adaptación al trabajo remoto y la evolución continua de los principios ágiles.

Marcano y Benigni (2014) evaluaron metodologías para el desarrollo de software educativo, considerando factores como el modelo de ciclo de vida (en cascada, evolutivo, incremental), la composición del equipo de trabajo, los requerimientos de documentación y la inclusión de etapas de prueba y validación. Por su parte, Parreira Júnior et al. (2009) examinaron la norma ISO/IEC/IEEE 12207 y concluyeron que esta ofrece una hoja de ruta para desarrollar o adquirir software de calidad que responda a las necesidades del usuario. Sin embargo, cumplir con todas las etapas prescritas resulta esencial para garantizar la calidad del software, lo que requiere un monitoreo continuo del proyecto. Irrazabal et al. (2011) exploraron la relación entre las prácticas ágiles—particularmente SCRUM—y un subconjunto de los procesos definidos por la norma ISO/IEC/IEEE 12207. Su estudio, basado en investigaciones previas y en la consulta a 25 empresas, evidenció la implementación de metodologías ágiles alineadas con los resultados esperados por el estándar. Cabe señalar que dicho análisis excluyó los procesos técnicos del estándar, concentrándose en los procesos de facilitación organizacional y gestión técnica.

Para alcanzar el objetivo del presente estudio, se llevó a cabo un ejercicio de mapeo de procesos que vinculó los procesos técnicos definidos por la norma ISO/IEC/IEEE 12207 con los procesos del ciclo de vida de cada metodología seleccionada. El análisis también identificó los métodos de ingeniería de software aplicados, el dominio de contenido de las aplicaciones objetivo, las validaciones realizadas sobre cada metodología y las técnicas de documentación empleadas. Este

análisis integral busca clarificar el grado de alineación de las metodologías multimedia con el estándar de desarrollo de software, así como evaluar el impacto de dichas prácticas en la calidad y efectividad del software educativo y comercial. En consecuencia, tanto estudiantes como desarrolladores pueden aplicar criterios científicos al momento de seleccionar la metodología de desarrollo multimedia más adecuada.

## **Materiales y Métodos**

Esta investigación adopta un diseño cualitativo, descriptivo y transversal. La metodología aplicada describe el método de recolección de datos, la estrategia analítica para evaluar las metodologías seleccionadas y una descripción detallada tanto de los procesos del estándar ISO/IEC/IEEE 12207 como de las metodologías de desarrollo de software analizadas.

### *Método de recolección de datos*

Se realizó una revisión de literatura utilizando bases de datos como Scopus, Emerald, Taylor & Francis, Google Scholar y EBSCOHost. La expresión de búsqueda “Multimedia software development methodology” se empleó para identificar las metodologías objetivo, centrándose en publicaciones comprendidas entre los años 1990 y 2025. Asimismo, se recuperaron fuentes adicionales mediante la expresión “Comparative study of software development methodologies”, con el fin de identificar investigaciones alineadas con los objetivos del presente estudio.

Una búsqueda inicial en Google Scholar utilizando la frase “Multimedia software development methodologies” arrojó 93,500 resultados. Una búsqueda avanzada refinada, limitada al periodo 1990–2025, redujo la cifra a 16,900 resultados. Un filtrado adicional, mediante la incorporación de la frase “Software life cycle”, produjo 3,870 resultados, lo que permitió identificar las metodologías más frecuentemente referenciadas.

### *Estrategia analítica de las metodologías*

El análisis metodológico tuvo como propósito responder las siguientes preguntas de investigación:

- ¿Qué procesos técnicos del estándar ISO 12207 corresponden a las etapas del ciclo de vida de las metodologías estudiadas?
- ¿Qué modelos de ingeniería de software se identifican en las metodologías analizadas?
- ¿Cuál es el dominio de aplicación (por ejemplo, educativo, comercial) al que se orienta cada metodología?
- ¿En qué medida (porcentaje) se alinean las metodologías seleccionadas con el estándar ISO/IEC/IEEE 12207?

Para cada metodología, los investigadores identificaron las fases del ciclo de vida y llevaron a cabo un ejercicio de mapeo de procesos con el fin de alinear dichas etapas con los procesos técnicos definidos en la norma ISO/IEC/IEEE 12207. La revisión de la literatura existente evidenció la ausencia de estudios previos que reporten el mapeo de metodologías de desarrollo de software multimedia frente a los procesos técnicos del estándar.

Este análisis se fundamenta en la experiencia metodológica presentada por Crisóstomo et al. (2017), quienes examinaron la relación entre dos modelos relevantes en la industria: ISO/IEC/IEEE 12207

(estándar de procesos del ciclo de vida del software) y CMMI-DEV (modelo de mejora de procesos). Su estudio confirmó que el mapeo de procesos constituye la técnica de comparación más ampliamente adoptada. Como herramienta sistemática y estructurada, el mapeo de procesos permite representar visualmente los flujos de trabajo, actividades, interacciones y dependencias, ofreciendo así una visión integral de los procesos, aclarando las relaciones entre etapas y revelando oportunidades para la optimización operativa. En la misma línea, Pardo et al. (2012) afirmaron que las técnicas de mapeo y comparación resultan útiles para armonizar modelos y estándares como CMMI, ISO 9001, ISO/IEC 15504 e ISO/IEC/IEEE 12207.

A partir de los enfoques de Baldassarre et al. (2009) y Crisóstomo et al. (2017), este estudio midió el grado en que cada metodología satisface los procesos técnicos definidos por la norma ISO/IEC/IEEE 12207. El porcentaje de cumplimiento se obtuvo mediante un análisis comparativo entre las etapas del ciclo de vida de cada metodología y los 14 procesos técnicos establecidos en el estándar.

El cálculo del porcentaje de cumplimiento siguió los siguientes pasos:

1. Mapeo de procesos: Identificación de las fases del ciclo de vida de cada metodología y su correspondencia con los procesos técnicos definidos en la norma ISO/IEC/IEEE 12207.
2. Escala de cobertura: Asignación de un porcentaje de cobertura a cada proceso técnico del estándar ISO, en función de su presencia y del grado de implementación dentro de la metodología:
  - Cobertura total (100 %): Todas las actividades y tareas del proceso técnico de la norma ISO/IEC/IEEE 12207 se encuentran completamente implementadas o explícitamente abordadas.
  - Cobertura parcial (50 %): Solo se implementaron o consideraron algunos aspectos o tareas de cada actividad del proceso técnico correspondiente del estándar ISO/IEC/IEEE 12207.
  - Cobertura mínima (25 %): Las tareas de las actividades de cada proceso técnico del estándar ISO/IEC/IEEE 12207 se consideraron de forma indirecta o solo fueron mencionadas.
  - Sin cobertura (0 %): Las tareas de las actividades de cada proceso técnico del estándar ISO/IEC/IEEE 12207 no fueron consideradas ni mencionadas.

Para calcular el promedio ponderado, el estudio sumó los porcentajes de cobertura correspondientes a los 14 procesos técnicos y dividió el total entre 14:

$$\text{Promedio} = (\sum \text{porcentajes de cobertura} / 14) \times 100$$

Para el análisis cualitativo, los investigadores emplearon la herramienta NotebookLM de Google, aplicando el siguiente prompt:

" En el documento del autor ..., titulado '...', se describe una metodología multimedia compuesta por múltiples fases. Aplica los criterios de mapeo de procesos del documento 'criterioAsignacionPorcentaje2025' y determina el porcentaje que debe asignarse a cada fase de la metodología según su alineación con los procesos técnicos definidos en la norma ISO/IEC/IEEE 12207. Presenta los resultados en una tabla que incluya los procesos técnicos, las fases relacionadas de la metodología y el porcentaje asignado."

NotebookLM recibió tres documentos de referencia: (1) el artículo científico que describe la metodología, (2) el documento *criterioAsignaciónPorcentaje2025.docx* que contiene la escala de cobertura definida, y (3) la norma ISO/IEC/IEEE 12207. Cada investigador revisó e interpretó los resultados de manera independiente.

NotebookLM, un asistente de investigación basado en inteligencia artificial, genera respuestas exclusivamente a partir de los documentos proporcionados por el usuario, con la ventaja añadida de identificar patrones semánticos. A diferencia de las herramientas tradicionales de análisis cualitativo—que requieren codificación y categorización manual—NotebookLM optimiza la detección de patrones y la generación de informes. Esta capacidad resulta esencial en la investigación cualitativa, donde la trazabilidad y la fidelidad a los datos originales constituyen aspectos críticos. Aunque NotebookLM actuó como herramienta de apoyo en este estudio, los investigadores conservaron la responsabilidad de interpretar los datos, asignar significados y formular los hallazgos clave.

#### Norma ISO/IEC/IEEE 12207 (ISO/IEC/IEEE, 2017)

La norma ISO/IEC/IEEE 12207 es un estándar internacional desarrollado conjuntamente por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC). Su título oficial es "Ingeniería de sistemas y software — Procesos del ciclo de vida del software", fue publicada por primera vez en 1995 y ha experimentado múltiples revisiones a lo largo de las décadas (ISO/IEC/IEEE, 2017).

Esta norma define un conjunto exhaustivo de procesos que abarca todo el ciclo de vida del software: desde la concepción y especificación del sistema, pasando por su implementación operativa y mantenimiento, hasta su retiro definitivo. Tres principios fundamentales sustentan estos procesos del ciclo de vida: cada proceso se vincula directamente con sus productos, actividades y tareas; los procesos operan de forma independiente entre sí; y cualquier proceso puede ejecutarse individualmente dentro del ciclo de vida. Cada proceso incluye un título, propósito, resultados esperados (outcomes), actividades y tareas.

La Tabla 1 presenta los procesos técnicos aplicables a la creación y uso de un sistema de software, ya sea como modelo o como producto operativo. Estos procesos técnicos resultan aplicables en todos los niveles jerárquicos de la arquitectura del sistema de software y en cualquier etapa del ciclo de vida.

Tabla 1. *Procesos técnicos*

<b>Proceso</b>	<b>Propósito</b>
Análisis de Negocio o Misión	Definir el problema, oportunidad de negocio o misión, caracterizar el espacio de la solución y determinar las posibles clases de soluciones que podrían abordar un problema o aprovechar una oportunidad.
Definición de Necesidades y Requerimientos de las Partes Interesadas	Definir los requisitos para un sistema que pueda proporcionar las capacidades que necesitan los usuarios en un entorno definido.
Definición de Requerimientos de Sistema/Software	Transformar la visión de las capacidades deseadas en una visión técnica de una solución que satisfaga las necesidades operativas del usuario.
Definición de Arquitectura	Generar alternativas de arquitectura del sistema, seleccionar alternativas que satisfagan los requisitos del sistema, y expresar esto en un conjunto de visiones consistentes.

Proceso	Propósito
Definición de Diseño	Proporcionar suficientes datos e información detallados sobre el sistema y sus elementos para permitir una implementación coherente con las entidades arquitectónicas definidas en los modelos y vistas de la arquitectura del sistema.
Análisis de Sistema	Proporcionar una base rigurosa de datos e información para la comprensión técnica que ayude a la toma de decisiones a lo largo del ciclo de vida. Se aplica al desarrollo de los datos de entrada necesarios para cualquier evaluación técnica. Puede brindar confianza en la utilidad e integridad de los requisitos, la arquitectura y el diseño del sistema.
Implementación	Realizar un elemento de sistema específico. Transforma los requisitos, la arquitectura y el diseño, incluidas las interfaces, en acciones que crean un elemento de sistema de acuerdo con las prácticas de la tecnología de implementación seleccionada.
Integración	Sintetizar un conjunto de elementos del sistema en un sistema realizado (producto o servicio) que satisfaga los requisitos, la arquitectura y el diseño del sistema/software. Ensambla los elementos del sistema implementados. Se identifican y activan las interfaces para permitir la interoperación de los elementos del sistema según lo previsto.
Verificación	Proporcionar evidencia objetiva de que un sistema o un elemento del sistema cumple con sus requisitos y características especificados. Identifica las anomalías (errores, defectos o fallas) en elementos de información, elementos del sistema implementados o procesos del ciclo de vida utilizando métodos, técnicas, estándares o reglas apropiados.
Transición	Establecer una capacidad para que un sistema proporcione servicios especificados por los requisitos de las partes interesadas en el entorno operativo.
Validación	Proporcionar evidencia objetiva de que el sistema, cuando está en uso, cumple con sus objetivos comerciales o de misión y los requisitos de las partes interesadas.
Operación	Utilizar el sistema para prestar sus servicios. Establece los requisitos y asigna personal para operar el sistema, y supervisa los servicios y el rendimiento del operador del sistema.
Mantenimiento	Mantener la capacidad del sistema para prestar un servicio. Supervisa la capacidad del sistema para prestar servicios, registra los incidentes para su análisis, toma medidas correctivas, adaptativas, perfeccionadoras y preventivas, y confirma la capacidad restaurada.
Desecho	Poner fin a la existencia de un elemento del sistema o de un sistema para un uso previsto específico, manejar apropiadamente los elementos reemplazados o retirados y atender apropiadamente las necesidades críticas de eliminación identificadas.

Fuente: Elaboración propia

## Methodologías

Tras el análisis de las metodologías recuperadas en los resultados de búsqueda, se seleccionaron ocho enfoques en función de su alineación con el objeto de estudio: el desarrollo de software multimedia. Estas metodologías fueron elegidas por su capacidad para definir fases de desarrollo diferenciadas, lo que permitió una comparación estructurada con los procesos técnicos establecidos en la norma ISO/IEC/IEEE 12207. A continuación, se presenta la descripción de cada metodología, ordenadas cronológicamente según su año de publicación.

### *Método para el Desarrollo de Aplicaciones Hipermedia Educativas (MDAEM)*

Este método integra principios de ingeniería de software y análisis cognitivo para asegurar tanto la funcionalidad educativa como la calidad técnica de las aplicaciones multimedia. Su ciclo de vida evolutivo permite ajustes continuos mediante evaluaciones iterativas, mejorando así la efectividad pedagógica y tecnológica. El enfoque pone énfasis en la calidad del producto final y en la satisfacción del usuario, garantizando que la aplicación cumpla con las expectativas funcionales y educativas (Valencia, 1998).

*Metodología para el Desarrollo de Productos y Sistemas Multimedia (MDPSM)*

Esta metodología ofrece un marco estructurado pero flexible, capaz de adaptarse a la evolución tecnológica. Da prioridad a la gestión de proyectos, aseguramiento de la calidad y gestión de riesgos. Su objetivo principal consiste en mejorar la calidad y eficiencia del diseño, producción y despliegue de productos multimedia, mediante la incorporación de estándares como ISO 9001 para garantizar la consistencia y optimización de procesos. Las actividades se agrupan en tres categorías—desarrollo, gestión y soporte—ofreciendo así una visión integral del ciclo de vida del producto (Sherwood & Rout, 1998).

*Metodología Orientada a Objetos para el Desarrollo de Software Multimedia e Hipermedia (MOOMH)*

MOOMH se centra en el desarrollo de sistemas multimedia e hipermedia mediante técnicas orientadas a objetos y principios de ingeniería de software. Su estructura se compone de tres fases principales: análisis, diseño e implementación. Incluye mecanismos de gestión y control de proyectos, respaldados por una herramienta CASE que optimiza la eficiencia del diseño. La metodología busca equilibrar el rigor técnico con la usabilidad, facilitando así el desarrollo de proyectos multimedia con fines educativos e informativos (Benigni, 2004).

*Modelo de Desarrollo de Software Educativo Basado en Competencias (MODESEC)*

Este modelo orienta la creación de software educativo desde un enfoque basado en competencias, integrando principios pedagógicos, de aprendizaje y de ingeniería de software. Estructura el proceso de desarrollo en cinco fases: diseño instruccional, diseño multimedia, diseño computacional, producción y despliegue. La metodología asegura la alineación entre el producto de software y las competencias específicas definidas en el proceso de enseñanza-aprendizaje (Caro Piñeres et al., 2009).

*Metodología para el Desarrollo de Objetos de Aprendizaje Multimedia (MESOVA)*

MESOVA se enfoca en el desarrollo de objetos de aprendizaje virtuales mediante un enfoque modular y evolutivo. El proceso inicia con la concepción del objeto, donde se definen los objetivos pedagógicos y los requisitos. Durante la fase de diseño y desarrollo modular, se crean módulos de aprendizaje de forma progresiva, los cuales se integran y validan en un entorno final. En la etapa de pruebas, se evalúa el impacto del objeto sobre el aprendizaje comparando grupos de control y experimentales. La fase de consolidación incluye el despliegue del objeto y su documentación, seguido de una evaluación continua para futuras mejoras (Parra Castrillón, 2011).

*Metodología para el Desarrollo de Software Educativo Multimedia (MEDESME)*

Esta metodología híbrida combina diversos enfoques de desarrollo de software—incluyendo modelos en espiral, incremental, en cascada, orientado a la reutilización y evolutivo—para asegurar software multimedia educativo de alta calidad. El proceso se desarrolla en ocho fases: concepto o preproducción, análisis de objetivos educativos y requisitos técnicos, diseño de interfaces y de componentes multimedia, desarrollo del software, integración de elementos multimedia, evaluación pedagógica y validación, producción y distribución de la versión final, y finalmente, elaboración de materiales complementarios como manuales de usuario y recursos didácticos. Esta estructura favorece un enfoque integral para el desarrollo de aplicaciones educativas multimedia eficaces (García Sánchez et al., 2016).

### *Proyecto Multimedia*

Vaughan (2014, p. 1) define un proyecto multimedia como un vehículo de software que contiene mensajes y contenido que se visualizan mediante ordenador, pantalla de televisión, dispositivo móvil o teléfono inteligente. El desarrollo de proyectos multimedia se estructura en cuatro etapas principales: planificación y estimación de costos, diseño y producción, pruebas, y entrega. Estas etapas incluyen fases específicas como análisis, prepruebas, desarrollo de prototipos, desarrollo alfa, desarrollo beta y entrega final. A lo largo del proceso, los ciclos de retroalimentación incorporan pruebas y ajustes realizados por expertos o usuarios para perfeccionar el producto final (Vaughan, 2014).

## **Metodología de Ingeniería de Software Multimedia (MISM)**

El desarrollo de software multimedia implica la convergencia de dos dominios distintos: la ingeniería de software, centrada en la creación estructurada de productos digitales, y el diseño multimedia, enfocado en los elementos visuales y creativos. Muchas metodologías tradicionales de desarrollo no logran integrar adecuadamente las fases de producción multimedia.

Con el fin de cerrar esta brecha, los autores proponen una metodología interdisciplinaria que combine el ciclo de vida del desarrollo de software con el proceso de producción multimedia. Este enfoque busca asegurar la calidad del producto mediante la integración eficiente de los aspectos técnicos y creativos a lo largo de todo el proyecto de desarrollo multimedia (Al-Jabari et al., 2019).

## **Resultados**

En términos generales, las etapas del ciclo de vida de un sistema de software comprenden la exploración del concepto, el desarrollo, el mantenimiento y el retiro, con transiciones entre cada una de ellas. Cada etapa involucra actividades de planificación, ejecución y evaluación. La secuencia en la que se desarrollan estas etapas depende del modelo de ciclo de vida adoptado. Entre los modelos más comúnmente aplicados se encuentran los secuenciales—como el modelo en cascada—y los no secuenciales—como los modelos incremental, en espiral, iterativo y evolutivo. Los modelos no secuenciales suelen incorporar técnicas y métodos ágiles (Washizaki, 2024).

Cabe señalar que, según el SWEBOK V4.0 editado por Washizaki (2024), los métodos de ingeniería de software pueden clasificarse, según su enfoque y aplicación, en las siguientes categorías: métodos heurísticos, métodos formales, métodos de prototipado y métodos ágiles.

La Tabla 2 identifica las etapas del ciclo de vida y el modelo correspondiente adoptado por cada una de las metodologías estudiadas—específicamente, si siguen un orden secuencial o no secuencial en la aplicación de las fases del desarrollo de software multimedia. Las metodologías basadas en modelos no secuenciales ponen énfasis en la validación continua y la adaptabilidad a los entornos multimedia.

**Tabla 2**

*Etapas y modelo de ciclo de vida de las metodologías estudiadas.*

<b>Metodología</b>	<b>Etapas del ciclo de vida</b>	<b>Modelo de ciclo de vida</b>
MDAEM (Valencia, 1998)	1 Descripción del problema, 2 Definición de requerimientos y términos del contrato 3 Diseño educativo 4 Diseño computacional	Evolutivo

Metodología	Etapas del ciclo de vida	Modelo de ciclo de vida
	5 Producción 6 Evaluación del producto.	
MDPSM (Sherwood & Rout, 1998)	1 Iniciación 2 Especificaciones 3 Diseño 4 Producción 5 Revisión y Evaluación 6 Entrega e Implementación	Iterativo, prototipado
MOOMH (Benigni, 2004)	1 Modelo de Análisis 2 Modelo de Diseño 3 Modelo de Implantación	Iterativo e incremental
MODESEC (Caro Piñeres et al., 2009)	1 Diseño Educativo 2 Diseño Multimedial 3 Diseño Computacional 4 Producción 5 Aplicación	Lineal solapado
MESOVA (Parra Castrillón, 2011)	1 Concepción del objeto 2 Diseño y desarrollo modular evolutivo 3 Integración y despliegue 4 Pruebas de aprendizaje 5 Consolidación 6 Evaluación y mejora continua	Híbrido
Proyecto Multimedia (Vaughan, 2014)	1 Planeamiento y costo 2 Diseño y producción 3 Prueba 4 Entrega	Ágil, Prototipado
MEDESME (García Sánchez et al., 2016)	1 Concepto o pre-producción 2 Análisis 3 Diseño 4 Desarrollo 5 Implementación 6 Evaluación y validación del programa 7 Producción 8 Elaboración de material complementario	Iterativo
MISM (Al-Jabari et al., 2019)	1 Requerimiento y Preproducción 2 Diseño y Producción 3 Validación y Postproducción 4 Evolución	No especificado

Fuente: Elaboración propia

Como se muestra en la Tabla 2, todas las metodologías analizadas adoptan un modelo de ciclo de vida no secuencial.

Las etapas del ciclo de vida listadas en la Tabla 2 indican que la mitad de las metodologías fueron formuladas específicamente para el desarrollo de aplicaciones multimedia en el sector educativo (Caro Piñeres et al., 2009; García Sánchez et al., 2016; Parra Castrillón, 2011; Valencia, 1998), mientras que la otra mitad se orienta al desarrollo de software multimedia en general (Al-Jabari et al., 2019; Benigni, 2004; Sherwood & Rout, 1998; Vaughan, 2014). La Tabla 5 presenta detalles adicionales sobre las principales etapas.

Con base en los hallazgos de Baldassarre et al. (2009) y Crisóstomo et al. (2017), se determinó el grado de alineación de cada metodología estudiada con los procesos técnicos definidos por la norma

ISO/IEC/IEEE 12207. La Tabla 3 resume los resultados del mapeo de procesos entre los procesos técnicos del estándar ISO/IEC/IEEE 12207 y las metodologías de desarrollo de software seleccionadas. Este mapeo funciona como una herramienta clave para comparar el estándar con las metodologías de desarrollo, al alinear sistemáticamente sus áreas de proceso. Tal alineación permite que las organizaciones evalúen la compatibilidad e integración de las metodologías con el estándar, en particular en lo referido a los procesos técnicos identificados.

La Tabla 3 revela que los procesos técnicos con mayor cobertura incluyen: Definición de Necesidades y Requisitos de las Partes Interesadas, Definición de la Arquitectura, Definición del Diseño e Implementación. En cambio, los procesos con menor cobertura—Operación, Mantenimiento y Disposición Final—no son abordados por ninguna de las metodologías. Los procesos de Verificación y Validación, que se refieren respectivamente a la evaluación durante el desarrollo y al cumplimiento de requisitos durante el uso del sistema, presentan niveles moderados de cobertura en las metodologías analizadas.

**Tabla 3**

*Porcentaje de cobertura de los procesos técnicos del estándar ISO/IEC/IEEE 12207 por metodología de estudio*

PROCESO TÉCNICO ISO/IEC/IEEE 12207	METODOLOGÍA															
	MDAEM (Valencia, 1998)		MDPSM (Sherwood & Rout, 1998)		MOOMH (Benigni, 2004)		MODESEC (Caro Piñeres et al., 2009)		MESOVA (Parra Castrillón, 2011)		Proyecto Multimedia (Vaughan, 2014)		MEDESME (García Sánchez et al., 2016)		MISM (Al-Jabari et al., 2019)	
	FASE <sup>a</sup>	P.C. <sup>b</sup>	FASE	P.C.	FASE	P.C.	FASE	P.C.	FASE	P.C.	FASE	P.C.	FASE	P.C.	FASE	P.C.
Análisis de Negocio o Misión	1	50%	1	50%	1	50%	1	50%	1	50%	1	50%	1	50%	1	50%
Definición de Necesidades y Requerimientos de las Partes Interesadas	1, 2	50%	2	100%	1	50%	1, 3	50%	1	50%	1, 2, 3	50%	2	100%	1	50%
Definición de Requerimientos de Sistema/Software	2	50%	2	100%	1	50%	1, 3	50%	1	50%	1	25%	2	100%	1	50%
Definición de Arquitectura	3, 4	50%	3	100%	2	50%	3	50%	1	50%	2	50%	3	100%	2	50%
Definición de Diseño	4	50%	3	100%	2	100%	2, 3	50%	1, 2	50%	2	50%	3	100%	2	50%
Análisis de Sistema	3	25%	3	50%	1	25%	3	50%	1	50%	1	50%	6	50%	1	50%
Implementación	5	50%	4	100%	3	50%	4	100%	1, 2	50%	2	50%	4	100%	2	50%
Integración	5	50%	4	100%	NA	0%	4	100%	3	50%	2	50%	NA	0%	2	50%
Verificación	6	50%	5	50%	3	50%	4	50%	1, 2, 3	50%	3	50%	6	100%	NA	0%
Transición	6	50%	6	50%	3	50%	5	50%	3, 5	50%	4	50%	5	50%	NA	0%
Validación	NA	0%	5, 6	100%	3	25%	5	50%	1, 2, 3, 4	50%	1, 3	50%	6	100%	3	50%
Operación	NA	0%	6	25%	NA	25%	5	50%	5	25%	NA	0%	7,8	25%	NA	0%
Mantenimiento	6	25%	5, 6	25%	NA	0%	4	25%	5	50%	4	25%	NA	0%	4	50%
Desecho	NA	0%	NA	0%	NA	0%	NA	0%	NA	0%	NA	0%	NA	0%	NA	0%
<b>Promedio ponderado</b>	<b>35%</b>		<b>67%</b>		<b>37%</b>		<b>51%</b>		<b>44%</b>		<b>39%</b>		<b>62 %</b>		<b>35%</b>	

Nota: a) El número de fase corresponde a las listadas en la Tabla 2. b) P.C. es equivalente a porcentaje de cobertura. c)NA indica que ninguna fase de la metodología contempla al proceso técnico

Fuente: Elaboración propia

La Tabla 4 resume la cobertura total o parcial de los procesos técnicos en las distintas metodologías. Con base en las Tablas 3 y 4, la metodología con el mayor porcentaje de cobertura corresponde a MDPSM, propuesta por Sherwood & Rout (1998), seguida por MEDESME, desarrollada por García Sánchez et al. (2016). La primera se enfoca en el desarrollo de productos multimedia de propósito general, mientras que la segunda está orientada al desarrollo de software educativo multimedia.

**Tabla 4**

*Análisis de mapeo de procesos técnicos*

Metodología	Porcentaje	Observación
MDAEM (Valencia, 1998)	35	Cubre parcialmente 9 de los 14 procesos técnicos del estándar. No contempla 3 de los 14 procesos. Esta metodología fue desarrollada antes de la consolidación del estándar.
MDPSM (Sherwood & Rout, 1998)	67	Cubre totalmente 7 de los 14 procesos técnicos del estándar. Cubre parcialmente procesos técnicos en el área de análisis de negocio, análisis de sistema, verificación y transición. Su principal debilidad está en operación, mantenimiento y desecho.
MOOMH (Benigni, 2004)	37	Solo cubre totalmente el proceso de definición de diseño. Cubre parcialmente 7 de los 14 procesos técnicos.
MODESEC (Caro Piñeres et al., 2009)	51	Cubre totalmente 2 de los 14 procesos, cubre parcialmente 10 de los 14 procesos técnicos.
MESOVA (Parra Castrillón, 2011)	44	Cubre parcialmente 12 de los 14 procesos técnicos del estándar. De manera mínima atiende el proceso de operación.
Proyecto Multimedia (Vaughan, 2014)	39	Cubre parcialmente 10 de los 14 procesos técnicos del estándar. De manera mínima atiende el proceso de definición de requerimientos y mantenimiento.
MEDESME (García Sánchez et al., 2016)	62	Cubre totalmente 7 de los 14 procesos técnicos del estándar, otros 3 procesos son cubiertos parcialmente.
MISM (Al-Jabari et al., 2019)	35	Cubre parcialmente 10 de los 14 procesos técnicos del estándar, los otros cuatro están ausentes.

Fuente: Elaboración propia

La Tabla 5 destaca elementos específicos de las metodologías de desarrollo de software multimedia que no se encuentran explícitamente presentes en enfoques de desarrollo de software más generales (por ejemplo, XP, RUP, cascada, espiral). Estos elementos presentan una mejor cobertura según el análisis de mapeo de procesos.

**Tabla 5**

*Análisis de los procesos técnicos con presencia en todas las metodologías de estudio*

Metodología	Definición de necesidades y requerimientos	Definición de arquitectura	Definición de diseño	Implementación
MDAEM (Valencia, 1998)	Análisis del contenido educativo y los objetivos pedagógicos del	No se menciona explícitamente, pero se puede inferir en la	Diseño de la interfaz y la navegación hipermedial, con énfasis en la	Creación del software utilizando herramientas

Metodología	Definición de necesidades y requerimientos	Definición de arquitectura	Definición de diseño	Implementación
	software. Incluye la identificación de necesidades del usuario y los recursos disponibles.	organización del contenido y su estructura hipermedial.	interacción humano-máquina.	hipermediales, con pruebas iniciales para verificar funcionalidad.
MDPSM (Sherwood & Rout, 1998)	Análisis del público objetivo y sus necesidades en términos multimedia.	Mapas de navegación y storyboards. Identificación de actividades humanas que soportará el sistema interactivo.	Integración de diseño de interfaz, navegación y formatos multimedia. Control de versiones. Aplicación de estándares de calidad para asegurar calidad y mejora continua.	Control riguroso de cambios en medios audiovisuales y gráficos, no solo en código. Pruebas de integración multimedia para garantizar compatibilidad entre elementos.
MOOMH (Benigni, 2004)	Modelo de análisis, donde se determinan las necesidades del sistema, los objetivos del proyecto y los actores involucrados.	Grafo de navegación del sistema.	Modelo de diseño, especifica cada nodo del grafo y los elementos multimedia asociados.	Incluye el desarrollo técnico del sistema basado en los modelos previos, utilizando herramientas como UML para aplicaciones web.
MODESEC (Caro Piñeres et al., 2009)	En la fase diseño educativo, se estudian los factores educativos y se definen los objetivos de aprendizaje y competencias a desarrollar.	En la fase multimedial se aborda la estética y el sistema de comunicación.	La fase diseño computacional incluye la elaboración de especificaciones del software y planificación de la implementación técnica.	En la fase producción se lleva a cabo la codificación, integración y validación del software desarrollado.
MESOVA (Parra Castrillón, 2011)	En la fase concepción del objeto se realizan actividades de caracterización de la temática y nivel educativo, especificación de requisitos funcionales y no funcionales	En la fase de diseño se elabora un perfil gráfico del objeto que visualiza los módulos constitutivos o subsistemas.	En la fase diseño <sup>1</sup> , el diseño modular global y los modelos de UML para describir interacciones y transiciones de estado se vinculan al diseño del software.	La fase desarrollo consiste en la codificación y los mapeos necesarios para crear los módulos del objeto de aprendizaje.
Proyecto Multimedia (Vaughan, 2014)	“Look and feel” creativo Mapas de contenido Cronograma de desarrollo de elementos multimedia	Estructura y sistema de navegación	Storyboards Producción multimedia (gráficos, sonidos, videos)	Sistema de autoría
MEDESME (García Sánchez et al., 2016)	Identificación de elementos multimedia. Estrategias de aprendizaje y tareas interactivas.	Diseño de la navegación: mapas de navegación, storyboards,	Diseño instruccional basado en modelos pedagógicos.	Ensamblaje de elementos multimedia y su integración en la interfaz.

Metodología	Definición de necesidades y requerimientos	Definición de arquitectura	Definición de diseño	Implementación
	Perfil del usuario: nivel educativo, conocimientos previos, actitud ante la tecnología. Entorno de aprendizaje.	diagramas de análisis de tareas.	Integración de la interfaz gráfica con los objetivos de aprendizaje. Selección de herramientas para edición multimedia	
MISM (Al-Jabari et al., 2019)	Estudio del dominio de la aplicación	Es parte del diseño de componentes, se basa en teorías y conjunto de reglas de diseño de software	Diseño y creación de componentes multimedia Diseño y desarrollo componentes de software (arquitectónico, interfaz)	Integración de componentes

Fuente: Elaboración propia

Otro elemento común observado en las metodologías estudiadas es la necesidad de contar con equipos interdisciplinarios para asegurar el éxito del proyecto. Entre los profesionales requeridos se encuentran desarrolladores de software, diseñadores gráficos, productores de audio y video, diseñadores instruccionales y gestores de proyecto.

En términos generales, las metodologías propuestas se fundamentan en revisiones de literatura, basadas en enfoques existentes de desarrollo de software. Los autores justifican sus propuestas a partir de las debilidades identificadas en metodologías previamente establecidas. En la mayoría de los casos, dichas metodologías han sido validadas mediante su aplicación práctica en el desarrollo de productos de software multimedia dentro del ámbito universitario (véase Tabla 6).

Las metodologías de Sherwood & Rout (1998) y Al-Jabari et al. (2019), por ejemplo, fueron revisadas y probadas por expertos involucrados en el desarrollo de aplicaciones multimedia. En cambio, la propuesta de Vaughan (2014) se origina en su experiencia profesional en la producción de contenidos multimedia.

Tabla 6. *Fundamentación y validación de las metodologías objeto de estudio*

Metodología	Fundamentación	Validación
MDAEM (Valencia, 1998)	Ingeniería de software y la norma ISO 9000-3	Se aplica para el desarrollo de aplicaciones educativas hipermedia en la Facultad de Ingeniería de la Universidad del Valle, en Cali, Colombia.
MDPSM (Sherwood & Rout, 1998)	Revisión de literatura, influencia de los estándares ISO/IEC 12207 y ISO/IEC 15504	Revisión por pares, prueba con grupos de estudiantes en el desarrollo de proyectos en la Licenciatura en Multimedia de la Universidad de Griffith.
MOOMH (Benigni, 2004)	Revisión de literatura de Ingeniería de software orientada a objetos y metodologías de desarrollo de software educativo	Se aplica para el desarrollo de aplicaciones en la Universidad de Oriente-Núcleo Nueva Esparta.
MODESEC (Caro Piñeres et al., 2009)	Se fundamenta en el sistema de competencias y procesos de desarrollo de software educativo	Aplicado en trabajos de grado y de asignaturas en el área de Desarrollo de Software Educativo

Metodología	Fundamentación	Validación
MESOVA (Parra Castrillón, 2011)	Revisión de literatura con elementos de XP, RUP, UP y de ciclo de vida en espiral, incremental y evolutivo, desarrollo ágil de prototipos.	dirigidos por el grupo de investigación Edupmedia. Se aplicó a la construcción de 13 objetos de aprendizaje para el área de la programación de computadoras, en la Fundación Universitaria Católica del Norte.
Proyecto Multimedia (Vaughan, 2014)	Se fundamenta en la gestión de proyectos	Se aplica en la industria del software multimedia
MEDESME (García Sánchez et al., 2016)	Revisión de literatura de ingeniería de software.	Aplicada en la producción de software educativo
MISM (Al-Jabari et al., 2019)	Se fundamenta en las metodologías de ingeniería de software y las fases de producción multimedia.	Comparación y evaluación con el estándar ISO/IEC/IEEE 12207:2017, revisión por pares expertos, aplicación en proyectos de estudiantes en la Universidad de Hebron.

Fuente: Elaboración propia

Es importante señalar que el SWEBOK V4.0, editado por Washizaki (2024), identifica dos tipos de modelado de software: el modelado estructural (composición física o lógica) y el modelado del comportamiento (funciones). Para respaldar esta distinción, el Lenguaje Unificado de Modelado (UML) proporciona un conjunto amplio de diagramas.

En este sentido, la Tabla 7 resume las principales técnicas y herramientas recomendadas para documentar el proceso de desarrollo de software multimedia. Por lo general, el componente estructural del software multimedia se documenta mediante mapas de navegación, bocetos de interfaces y storyboards. En cambio, el componente orientado al comportamiento—impulsado por eventos—se modela utilizando diagramas UML como los diagramas de secuencia y los diagramas de casos de uso.

Asimismo, las herramientas y métodos empleados para las actividades de desarrollo pueden variar en función del tipo de producto que se desea crear. Por ejemplo, en el desarrollo de productos multimedia, la generación y el análisis de ideas suelen apoyarse en storyboards, esquemas y bocetos. En contraste, para productos de software más generales, estas actividades suelen basarse en entrevistas, observación y descripciones de casos de uso.

**Tabla 7.** Modelado de software

Metodología	Técnicas de documentación sugeridas
MDAEM (Valencia, 1998)	Establece la elaboración de documentos finales para cada fase, pero no sugiere ninguna técnica de documentación específica. Entre los documentos sugeridos están: diseño educativo con el análisis de tareas de aprendizaje, diseño del formato patrón de las unidades de información, especificación estética de los ítems de información.
MDPSM (Sherwood & Rout, 1998)	Cuenta con documentación en función a cada fase y las actividades que se realizan en ellas. Como el plan del proyecto, especificaciones funcionales, mapas de navegación, plantillas de pantallas, storyboard y flujogramas, estrategia de implementación de evaluación, manuales.
MOOMH (Benigni, 2004)	Especificación de requerimientos, elaboración de mapa de navegación, diseño de lecciones o unidades de información, grafo de navegación, boceto de las unidades de información, biblioteca de objetos multimedia, manual de usuario.

Metodología	Técnicas de documentación sugeridas
MODESEC (Caro Piñeres et al., 2009)	Formato de diseño de competencias, manual de usuario, listado de conceptos, matriz para diseño de contenidos, diagramas de contenido, mapa de navegación y descripción de interfaces. Diagramas de casos de uso, diagramas de clases, diagramas de objetos, diagramas de secuencia, y modelos de bases de datos como MER (Modelo Entidad Relación), MR (Modelo Relacional)
MESOVA (Parra Castrillón, 2011)	Resumen ejecutivo, manual de usuario y plan de soporte. Documentación formal concisa y navegable.
Proyecto Multimedia (Vaughan, 2014)	Describe los elementos a considerar en la presentación de la propuesta del proyecto: análisis de necesidades, descripción de la población destino, estrategia creativa, modelo de implementación del proyecto. El diseño multimedia interactivo se puede representar mediante <i>storyboard</i> usando un <i>wireframing</i> o una guía esquemática. El diseño de la estructura se puede presentar mediante diagramas de navegación (lineales, jerárquicos, no-lineales, compuestos).
MEDESME (García Sánchez et al., 2016)	Documentación de los requerimientos y análisis, ficha general y técnico psicopedagógica del diseño, ficha técnica de los requerimientos de software, esquema de navegación, mapa de navegación, <i>storyboard</i> , diagrama de análisis de tareas, diagramas de casos de uso y diagrama de relación de entidad, diseño de plantillas de la interfaz. Se generan manuales técnicos, de usuario y de actividades didácticas.
MISM (Al-Jabari et al., 2019)	Estudio de factibilidad, planeamiento del proyecto, análisis de riesgo. En las etapas de diseño y producción, sugiere el modelado de componentes, diseño arquitectónico, diseño de interfaces, sin especificar técnicas de documentación particulares.

Fuente: Elaboración propia

## Discusión

Las metodologías clásicas de desarrollo de software, aunque ampliamente adoptadas, resultan insuficientes para abordar los desafíos específicos que plantea el desarrollo de aplicaciones multimedia—especialmente en contextos educativos, como lo señalan Al-Jabari et al. (2019). Este hallazgo refuerza la premisa de que las aplicaciones multimedia requieren enfoques metodológicos diferenciados, capaces de gestionar la complejidad inherente a la integración de múltiples tipos de datos, facilitar la interacción usuario-sistema y la experiencia del usuario, y reconocer el papel del software multimedia como herramienta pedagógica. Estas conclusiones coinciden con lo planteado por Marcano y Benigni (2014), quienes destacaron las características particulares del desarrollo de software educativo.

No obstante, algunas metodologías demuestran aplicabilidad tanto en contextos educativos como comerciales, lo que subraya su versatilidad y capacidad de adaptación a diversos entornos.

La mayoría de las metodologías analizadas adoptan modelos de ciclo de vida no secuenciales—como los enfoques iterativo, evolutivo, ágil o híbrido. Esta tendencia refleja una creciente valorización de la flexibilidad, la adaptabilidad y la retroalimentación continua en el desarrollo de software multimedia, hallazgo consistente con lo señalado por Tetteh (2024), quien destacó la relevancia de las metodologías ágiles en la industria del software contemporánea.

El mapeo entre las fases de las metodologías y los procesos técnicos establecidos en la norma ISO/IEC/IEEE 12207 evidenció brechas significativas y áreas susceptibles de mejora. En la mayoría de los casos, la alineación con el estándar resulta parcial, con deficiencias notables en los

procesos de definición de requisitos, diseño, implementación y documentación. Cabe resaltar que ninguna de las metodologías examinadas contempla los procesos de mantenimiento y disposición, a pesar de su importancia crítica dentro del estándar.

Esta omisión podría afectar negativamente la calidad y sostenibilidad de los productos multimedia, lo que pone de relieve la necesidad de adaptar o diseñar metodologías que integren explícitamente estos procesos normativos. Este resultado refuerza lo planteado por Irrazabal et al. (2011), quienes destacaron la importancia de evaluar la relación entre las metodologías de desarrollo y la norma ISO/IEC/IEEE 12207, aunque su estudio se centró en las prácticas ágiles y los procesos de gestión técnica.

Desde una perspectiva práctica, los hallazgos sugieren que la selección o diseño de metodologías para el desarrollo multimedia debe incorporar prácticas que garanticen el cumplimiento de procesos técnicos esenciales, conforme a los estándares internacionales. Esta conclusión coincide con los aportes de Parreira Júnior et al. (2009), quienes señalaron la norma ISO/IEC/IEEE 12207 como un marco clave para orientar el desarrollo de software de alta calidad.

El estudio se basó en el análisis documental y el mapeo teórico de procesos, sin validación empírica directa en proyectos reales. Esta limitación restringe la generalización de los resultados y pone de manifiesto la necesidad de desarrollar investigaciones futuras que incluyan estudios de caso y validaciones empíricas en entornos reales de desarrollo multimedia.

## **Conclusiones**

A través del mapeo sistemático de las fases de las metodologías analizadas en relación con los procesos técnicos definidos por la norma ISO/IEC/IEEE 12207, se evidencia una brecha técnica: todas las metodologías presentan una cobertura limitada de procesos críticos del estándar, tales como operación, mantenimiento, verificación y validación. Solo las metodologías MDPSM, MODESEC y MEDESME cubren más del 50 % de los procesos técnicos requeridos por la ISO/IEC/IEEE 12207, seguidas por MESOVA con un 44 %, mientras que las restantes metodologías no superan el 40 %. Este patrón refleja un énfasis en los enfoques pedagógicos, evidenciado por una mayor cobertura de procesos relacionados con la definición de necesidades, el diseño instruccional y la implementación de contenidos, mientras que se descuidan aspectos técnicos esenciales.

El análisis comparativo reafirma la necesidad de adaptar o diseñar metodologías específicamente orientadas al desarrollo de software multimedia. Estos enfoques deben abordar no solo las características propias del dominio—como la interactividad, la usabilidad y la experiencia del usuario—sino también garantizar el cumplimiento de estándares internacionales como la norma ISO/IEC/IEEE 12207, en consonancia con las recomendaciones de Parreira Júnior et al. (2009). Para lograr dicha alineación, se recomienda integrar prácticas derivadas de ISO/IEC 15504 o CMMI-DEV (Crisóstomo et al., 2017).

En definitiva, esta investigación documenta las metodologías existentes para el desarrollo de software multimedia, respondiendo así a las necesidades identificadas por Al-Jabari et al. (2019). Además, sienta las bases para mejorar las metodologías actuales y diseñar alternativas más robustas y sensibles al contexto, alineadas con estándares internacionales, contribuyendo así a una mayor

calidad y efectividad en el desarrollo de software multimedia. El estudio también aborda el problema de generalización señalado por Hannington & Reed (2002).

Se recomienda que investigaciones futuras se enfoquen en el desarrollo de herramientas de apoyo para el mapeo automatizado de procesos conforme a los estándares pertinentes—una línea de trabajo coherente con Parreira Júnior et al. (2009)—así como en la evaluación del impacto de estas metodologías sobre la calidad de productos multimedia, tanto educativos como comerciales.

## Referencias

- Al-Jabari, M., Tamimi, T. K., & Ramadan, A.-A. N. (2019). Multimedia Software Engineering Methodology: A Systematic Discipline for Developing Integrated Multimedia and Software Products. *Software Engineering*, 8(1), 1-10. <https://doi.org/10.5923/j.se.20190801.01>
- Baldassarre, M. T., Piattini, M., Pino, F. J., & Visaggio, G. (2009). Comparing ISO/IEC 12207 and CMMI-DEV: Towards a mapping of ISO/IEC 15504-7. *2009 ICSE Workshop on Software Quality*, 59-64. <https://doi.org/10.1109/WOSQ.2009.5071558>
- Benigni, G. (2004). Una metodología orientada a objetos para la producción de software multimedia. *Saber, Universidad de Oriente, Venezuela*, 16(1), 26-32.
- Caro Piñeres, M. F., Toscazo Miranda, R. E., Hernández Roza, F. M., & David Lobo, M. E. (2009). Diseño de software educativo basado en competencias. *Ciencia e Ingeniería Neogranadina*, 19(1), 71-98.
- Crisóstomo, J., Melendez, K., Flores, L., & Dávila, A. (2017). Convergence Analysis of ISO/IEC 12207 and CMMI-DEV: Complementary Result from Systematic Literature Review. *CLEI Electronic Journal*, 20(3). <https://doi.org/10.19153/cleiej.20.3.7>
- Despa, M. L. (2014). Comparative study on software development methodologies. *Database Systems Journal*, V(3), 37-56.
- García Sánchez, E., Vite Chávez, O., Navarrate Sánchez, M. Á., García Sánchez, M. Á., & Torres Cosío, V. (2016). Metodología para el desarrollo de software multimedia educativo MEDESME. *CPU-e: Revista de Investigación Educativa*, 23, 216-226. <https://doi.org/10.25009/cpue.v0i23.2169>
- Gbaranwi, Precious, B., Ojekudo, & Akpofure, N. (2021). A Comparative Analysis of Software Development Methodologies. *International Journal of Research and Innovation in Applied Science*, 06(05), 159-166. <https://doi.org/10.51584/IJRIAS.2021.6513>
- Hannington, A., & Reed, K. (2002). Towards a taxonomy for guiding multimedia application development. *Ninth Asia-Pacific Software Engineering Conference, 2002.*, 97-106. <https://doi.org/10.1109/APSEC.2002.1182979>
- Irrazabal, E., Vásquez, F., Díaz, R., & Garzás, J. (2011). Applying ISO/IEC 12207:2008 with SCRUM and Agile Methods. En R. V. O'Connor, T. Rout, F. McCaffery, & A. Dorling (Eds.), *Software Process Improvement and Capability Determination* (Vol. 155, pp. 169-180). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-21233-8\\_15](https://doi.org/10.1007/978-3-642-21233-8_15)

- Islam, A. K. M. Z., & Ferworn, Dr. A. (2020). A Comparison between Agile and Traditional Software Development Methodologies. *Global Journal of Computer Science and Technology*, 20(2), 7-42. <https://doi.org/10.34257/GJCSTCVOL20IS2PG7>
- ISO/IEC/IEEE. (2017). *Systems and Software engineering-Software life cycle processes* (ISO/IEC/IEEE 12207:2017(E)).
- Marcano, I., & Benigni, G. (2014). Análisis de alternativas metodológicas para el desarrollo de software educativo. *Saber, Universidad de Oriente, Venezuela*, 26(3), 297-304.
- Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*, 14(4), 1504-1522. <https://doi.org/10.1007/s13198-023-01958-5>
- Mujumdar, A., Masiwal, G., & Chawan, P. M. (2012). Analysis of various Software Process Models. *International Journal of Engineering Research and Applications (IJERA)*, 2(3), 2015-2021.
- Pardo, C., Pino, F. J., García, F., Piattini, M., & Baldassarre, M. T. (2012). An ontology for the harmonization of multiple standards and models. *Computer Standards & Interfaces*, 34(1), 48-59. <https://doi.org/10.1016/j.csi.2011.05.005>
- Parra Castrillón, E. (2011). Propuesta de metodología de desarrollo de software para objetos virtuales de aprendizaje -MESOVA-. *Revista Virtual Universidad Católica del Norte*, 34, 113-137.
- Parreira Júnior, W. M., Ferreira Júnior, J. L. A., & Silva, L. P. da. (2009). UM ESTUDO DOS PROCESSOS DE CICLO DE VIDA DE SOFTWARE A PARTIR DA NORMA ISO 12207. *Intercursos Revista Científica*, 8(2), 167-176.
- Pressman, R. S. (2010). *Software engineering: A practitioner's approach* (7th ed). McGraw-Hill.
- Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition and The Standard for Project Management (ENGLISH)* (7th edition). Project Management Institute.
- Saleh, S. M., Rahman, M. A., & Asgor, K. A. (2017). Comparative Study on the Software Methodologies for Effective Software Development. *International Journal of Scientific & Engineering Research*, 8(4), 1018-1025.
- Sherwood, C., & Rout, T. (1998). A Structured Methodology for Multimedia Product and Systems Development. *ASCILITE*, 617-625.
- Tetteh, S. G. (2024). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *Asian Journal of Research in Computer Science*, 17(5), 30-42. <https://doi.org/10.9734/ajrcos/2024/v17i5436>
- Valencia, M. E. (1998). *Un método de desarrollo de aplicaciones educativas hipermedia*. Taller Internacional de Software Educativo TISE'97, Colombia.
- Vaughan, T. (2014). *Multimedia: Making it work* (Ninth edition). McGraw-Hill Education.
- Washizaki, H. (Ed.). (2024). *Guide to the Software Engineering Body of Knowledge v4.0: SWEBOK: Vol. v4.0*. IEEE Computer Society.