

Evaluación de la calidad del software en sistemas de educación superior utilizando el modelo CMMI
Evaluation of Software Quality in Higher Education Systems Using the CMMI Model
Ibeth Cedeño Delgado, Miguel Rodríguez Véliz .

Resumen

Este estudio analiza la implementación del Capability Maturity Model Integration (CMMI) en instituciones de educación superior para mejorar la calidad del software en aspectos clave como funcionalidad, accesibilidad y sostenibilidad. A través de un mapeo sistemático, se identificaron métricas como indicadores clave de desempeño (KPIs), satisfacción del usuario y cumplimiento de prácticas específicas, adaptadas a objetivos pedagógicos. La integración de CMMI con metodologías ágiles como Scrum demuestra su adaptabilidad a las necesidades dinámicas del entorno educativo. Los resultados destacan el potencial del CMMI para alinear la calidad técnica con los objetivos educativos, ofreciendo un marco estructurado para optimizar sistemas tecnológicos en instituciones con recursos limitados. Se recomiendan estudios empíricos futuros para validar su aplicabilidad.

Palabras claves: Calidad del software, Modelo Integrado de Madurez de Capacidades (CMMI), Sistemas de educación superior, Mejora de procesos, Metodologías ágiles, Accesibilidad y usabilidad.

Abstract

This study analyzes the implementation of the Capability Maturity Model Integration (CMMI) in higher education institutions to enhance software quality in key areas such as functionality, accessibility, and sustainability. Through a systematic mapping process, metrics were identified, including key performance indicators (KPIs), user satisfaction, and adherence to specific practices, all tailored to pedagogical objectives. The integration of CMMI with agile methodologies such as Scrum highlights its adaptability to the dynamic demands of the educational environment. The findings emphasize CMMI's potential to align technical quality with educational goals, providing a structured framework for optimizing technological systems in institutions with limited resources. Future empirical studies are recommended to validate its applicability.

Keywords: Software Quality, Capability Maturity Model Integration (CMMI), Process Improvement, Agile Methodologies, Accessibility, and Usability.

**INNOVACIÓN Y
CONVERGENCIA: IMPACTO
MULTIDISCIPLINAR**
Enero - Junio, V°6 - N°1;
2025

- ✓ **Recibido:** 30 /012/2024
- ✓ **Aceptado:** 14/01/2025
- ✓ **Publicado:** 31/01/2025

PAIS

- Ecuador, Portoviejo.
- Ecuador, Portoviejo

INSTITUCIÓN

- Universidad Técnica de Manabí
- Universidad Técnica de Manabí

CORREO:

- ✉ cedenoibeth09@gmail.com
- ✉ miguel.rodriguez@utm.edu.ec

ORCID:

- 🌐 <https://orcid.org/0009-0004-5772-3418>
- 🌐 <https://orcid.org/0000-0003-4474-3853>

FORMATO DE CITA APA.

Cedeño, I. Rodríguez, M. (2025). Evaluación de la calidad del software en sistemas de educación superior utilizando el modelo CMMI. Revista G-ner@ndo, V°6 (N°1.), 192 – 214.

Introducción

En el contexto actual, las instituciones de educación superior enfrentan crecientes desafíos para garantizar la calidad de sus sistemas tecnológicos debido a la acelerada evolución tecnológica y las diversas demandas de estudiantes, docentes y administradores. Estos sistemas no solo deben ser confiables y escalables, sino también adaptables a los objetivos pedagógicos y accesibles para usuarios con diferentes habilidades. La calidad del software es un factor crítico que impacta tanto en la experiencia de aprendizaje como en la eficiencia operativa de las instituciones educativas.

El Capability Maturity Model Integration (CMMI), desarrollado por el Software Engineering Institute (SEI), se ha consolidado como uno de los marcos más reconocidos para la evaluación y mejora de procesos de software. Este modelo, ampliamente aplicado en sectores industriales y empresariales, evalúa la madurez de los procesos en una escala progresiva, proporcionando un enfoque estructurado para optimizar recursos y garantizar la sostenibilidad operativa. Su flexibilidad permite su adaptación a contextos específicos, como el educativo, donde la integración de objetivos pedagógicos y tecnológicos es esencial para garantizar la efectividad y accesibilidad de los sistemas (Rojas Rodríguez & Torres López, 2019) (Piwowar-Sulej et al., 2022).

Este estudio analiza cómo el modelo CMMI puede ser implementado en instituciones de educación superior para evaluar y mejorar la calidad de sus sistemas tecnológicos. Se exploran enfoques metodológicos y métricas clave como la funcionalidad, accesibilidad, usabilidad y sostenibilidad. Además, se revisa la integración de CMMI con metodologías ágiles como Scrum para abordar los requisitos dinámicos del sector educativo. Este trabajo tiene como objetivo proporcionar un marco teórico que guíe a las instituciones en la optimización de sus sistemas tecnológicos, alineándolos con objetivos pedagógicos y mejorando la experiencia de usuarios finales (Aljedaibi & Talal Agal, 2022; Amer et al., 2020).

La evaluación de la calidad del software en instituciones de educación superior representa un desafío significativo debido a la creciente complejidad de los sistemas y las altas expectativas de calidad, a menudo enfrentadas con recursos limitados (Keshta, 2022; Khraiwesh, 2020). El Capability Maturity Model Integration (CMMI), desarrollado por el SEI, es ampliamente reconocido como un marco estructurado para mejorar procesos organizativos, evaluando la madurez en una escala de 1 a 5 e integrando disciplinas como sistemas y software (Piwowar-Sulej et al., 2022; Rojas Rodríguez & Torres López, 2019). Su implementación en sectores tecnológicos ha optimizado recursos y garantizado sostenibilidad operativa (Carvalho et al., 2020a). Este enfoque resulta esencial para plataformas de gestión académica y sistemas de aprendizaje robustos, que dependen de un software confiable y eficiente (Keshta, 2022).

Aunque el CMMI se destaca por su flexibilidad y capacidad de adaptación, otros modelos también han sido utilizados en el ámbito de la evaluación de software. Por ejemplo, el estándar ISO/IEC 9126 proporciona un marco para medir características como funcionalidad, confiabilidad y usabilidad. Sin embargo, tiene limitaciones en su capacidad de adaptación a contextos específicos, como los educativos, donde las necesidades pueden variar ampliamente (Ayyagari & Atoum, 2019; Laitinen et al., 2024).

Por su parte, el modelo SPICE (ISO/IEC 15504) ofrece un enfoque similar al CMMI al evaluar la madurez de procesos organizativos. No obstante, su énfasis está más orientado hacia la estandarización técnica, lo que puede limitar su aplicabilidad para evaluar software educativo en entornos con recursos limitados (Tharun Thomas & P.G. Saleeshya, 2024).

Otro modelo destacado es Six Sigma, ampliamente utilizado en la industria para reducir defectos y mejorar procesos. Aunque efectivo en entornos industriales, su aplicación en instituciones educativas es limitada, ya que no aborda específicamente dimensiones pedagógicas o de accesibilidad necesarias para sistemas académicos (Ayyagari & Atoum, 2019).

Dado que cada modelo presenta fortalezas particulares, también es importante analizar sus limitaciones en relación con las necesidades del sector educativo. En la Tabla 1 se presentan las principales limitaciones de tres modelos ampliamente utilizados en la evaluación de la calidad del software: ISO/IEC 9126, SPICE (ISO/IEC 15504) y Six Sigma.

Tabla 1. Principales limitaciones de los modelos ISO/IEC 9126, SPICE (ISO/IEC 15504) y Six Sigma en el contexto de la evaluación de software en centros educativos.

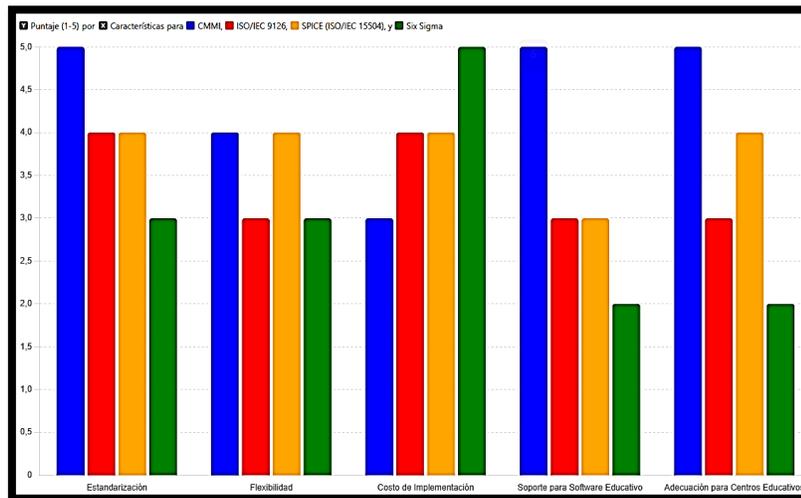
Modelo	Limitaciones Principales
ISO/IEC 9126	No se adapta a contextos educativos específicos; carece de métricas para accesibilidad y objetivos pedagógicos.
SPICE (ISO/IEC 15504)	Enfocado en estandarización técnica; costoso para instituciones con recursos limitados; no considera accesibilidad.
Six Sigma	Ignora dimensiones pedagógicas y accesibilidad; no es flexible ante cambios en requisitos educativos.

Para destacar las ventajas del modelo Capability Maturity Model Integration (CMMI) frente a otros marcos como ISO/IEC 9126, SPICE (ISO/IEC 15504) y Six Sigma, se presenta en la Figura 1 un análisis comparativo que evalúa dimensiones clave, como estandarización, flexibilidad, costos de implementación, soporte para software educativo y adecuación para centros educativos.

El gráfico que se presenta a continuación compara el modelo Capability Maturity Model Integration (CMMI) con otros marcos reconocidos, como ISO/IEC 9126, SPICE (ISO/IEC 15504) y Six Sigma, en términos de su capacidad para evaluar la calidad del software en instituciones educativas. La información utilizada para esta comparación se basa en el artículo de (Tharun Thomas & P.G. Saleeshya, 2024), quienes analizan la eficacia del modelo CMMI en diferentes contextos, destacando su aplicabilidad en la evaluación de procesos tecnológicos en entornos educativos. Este análisis también considera las conclusiones de (Alshazly et al., 2021; Ayyagari & Atoum, 2019; Oyshi et al., 2023), quienes abordan la simplificación del modelo CMMI para

organizaciones pequeñas y educativas, subrayando su flexibilidad y efectividad en recursos limitados.

Fig.1 Comparación de modelos de evaluación de calidad de software para centros educativos.



El modelo CMMI se posiciona como líder en varias métricas relevantes, especialmente en su adaptabilidad para contextos educativos, mientras que otros modelos, como Six Sigma, tienen un enfoque más limitado en términos de aplicaciones específicas para software educativo. Esto reafirma la capacidad del CMMI para integrar objetivos pedagógicos y tecnológicos, optimizando procesos de evaluación y mejora en instituciones educativas (Tharun Thomas & P.G. Saleeshya, 2024).

La integración de CMMI con metodologías ágiles, como Scrum, ha sido propuesta para aumentar la flexibilidad en contextos educativos (Elfaki & Bassfar, 2020) plantean un enfoque híbrido que combina agilidad con estándares de calidad, mientras que (Pries-Heje et al., 2023) destacan la importancia de identificar problemas en etapas tempranas mediante métricas específicas. Este último método facilita la trazabilidad y optimización de procesos, siendo clave para instituciones con recursos limitados.

Además de los enfoques orientados a procesos, se han abordado aspectos críticos como la seguridad.(Al-Matouq et al., 2020) proponen un modelo de madurez enfocado en prácticas seguras, destacando la formación organizacional como pilar para mejorar capacidades. Por su parte, (Yan et al., 2020) investigan la relación entre calidad del software y gestión de proyectos mediante redes neuronales, proponiendo modelos que optimizan recursos y garantizan estándares en sistemas académicos.

El uso combinado de modelos como TQM y CMMI ha mostrado ser efectivo para mejorar la satisfacción del cliente y la eficiencia organizativa, como sugieren (Jaiswal & Garg, 2019) . En términos de innovación, (Porto et al., 2021) analizan la gamificación en ingeniería de software, identificando beneficios como la motivación del equipo y la calidad del trabajo, aunque señalan retos como la resistencia al cambio. Adicionalmente, (Astridita et al., 2024) destacan beneficios como la precisión en estimaciones, planificación más efectiva y gestión optimizada de riesgos.

A pesar de los beneficios, la implementación de CMMI enfrenta desafíos significativos en el sector educativo, incluidos recursos limitados y la necesidad de personal capacitado (Carvalho et al., 2020b; Jadoon et al., 2020) . Finalmente, (Zarour & Alenezi, 2021) enfatizan la importancia del compromiso organizacional en la adopción del modelo, mientras que (Agrawal & Chari, 2020; Cerdeiral & Santos, 2019) subrayan el papel de la auditoría y el control en proyectos de alta madurez. Este estudio busca llenar la brecha existente al adaptar el CMMI para mejorar los sistemas en instituciones educativas (Keshta, 2022; Oyshi et al., 2023).

Método y materiales

La investigación documental es adecuada para el estudio “Evaluación de la Calidad del Software en Sistemas de Educación Superior utilizando el Modelo CMMI” porque permite analizar y compilar información existente sobre el modelo CMMI y su aplicación en contextos educativos y tecnológicos. Este tipo de investigación se basa en la revisión de literatura científica y estudios

de caso que documentan experiencias previas y mejores prácticas en la evaluación de la calidad del software. Al utilizar fuentes secundarias, se obtiene una base teórica sólida que resulta crucial para comprender cómo el modelo CMMI ha sido implementado y evaluado en distintos escenarios, incluyendo sistemas de educación superior. Esta información es esencial para identificar oportunidades y adaptar estos conocimientos al análisis de los sistemas tecnológicos utilizados en instituciones educativas.

Esta investigación se basa en una revisión sistemática de la literatura (RSL) sobre la evaluación de la calidad del software en sistemas de educación superior utilizando el modelo CMMI. La metodología sigue las pautas establecidas por (Barbara Kitchenham & Stuart M. Charters, 2007), lo que garantiza un enfoque estructurado y reproducible. El RSL se desarrolla en tres etapas principales: planificación, para definir los objetivos y criterios de inclusión; revisión, para identificar y seleccionar literatura relevante; y análisis, para extraer y sintetizar información clave que permita comprender el estado actual del uso del modelo CMMI en este contexto.

El método de investigación más adecuado para evaluar la calidad del software en sistemas de educación superior utilizando el modelo CMMI es una combinación de enfoques deductivo y analítico:

Deductivo: Este enfoque permite partir del marco teórico y las directrices establecidas por el modelo CMMI para aplicarlas al contexto de los sistemas de educación superior. Se utiliza la teoría existente para evaluar y analizar cómo las instituciones educativas implementan y adaptan el modelo CMMI en sus sistemas tecnológicos.

Analítico: Consiste en desglosar y analizar detalladamente los componentes clave de los sistemas tecnológicos de educación superior para identificar áreas de mejora, posibles incumplimientos con los estándares de calidad y soluciones específicas. Este enfoque proporciona un análisis exhaustivo que permite comprender las interacciones entre los elementos

del sistema y evaluar su alineación con los estándares de calidad propuestos por el modelo CMMI.

Para enmarcar la investigación sobre la problemática planteada, se plantean preguntas de investigación (PI) se establecieron, los cuales se definieron siguiendo los objetivos de este estudio:

RQ1: ¿Cuál es el estado actual del conocimiento sobre el modelo CMMI y su aplicación en la evaluación de la calidad del software?

RQ 2: ¿Qué métricas son más relevantes y efectivas para evaluar la calidad del software según el modelo CMMI en el contexto de sistemas de educación superior?

RQ3: ¿Cómo se traducen los principios y metodologías del modelo CMMI en acciones concretas para evaluar la calidad del software en sistemas de educación superior?

Para la selección de los estudios se tuvieron en cuenta los títulos y resúmenes correspondientes, cuenta en la que se anotarán los siguientes criterios de inclusión:

Estudios en el campo de la ciencia de la computación;

Artículos publicados desde 2019;

Artículos científicos de revistas.

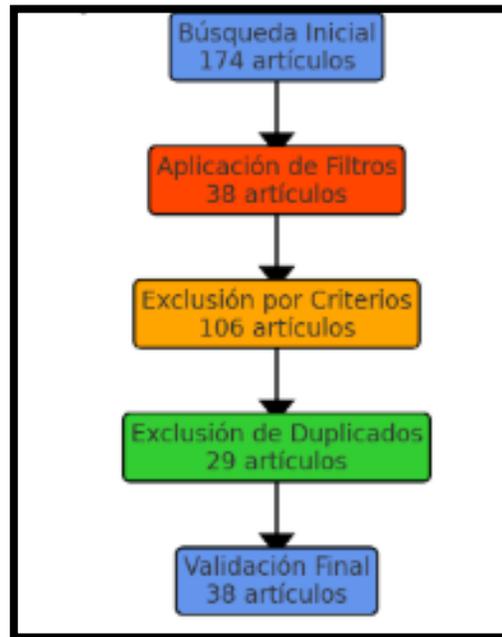
Dentro de los criterios de exclusión se consideraron los siguientes parámetros:

Estudios que no fueron incluidos en las bases de datos seleccionadas;

Estudios duplicados;

Estudios que no presentaron contenidos en ensayos en inglés o español.

Fig. 2 Diagrama de flujo del proceso de selección de artículos.



Se consideraron las siguientes bibliotecas digitales para la revisión de la literatura: IEEE Xplore, Researchgate, ScienceDirect y Scopus. La razón para seleccionarlos es porque cubren una amplia gama de publicaciones en el campo de las ciencias de la ingeniería manteniendo una base de datos completa y consistente. Para realizar la búsqueda inicial utilizamos palabras clave y términos de búsqueda.

Software Quality

Capability Maturity Model Integration

evaluation of software quality

Quality Management

Después de analizar diversas opciones, se evaluaron criterios que se alinean con los términos mencionados anteriormente para determinar la cadena de búsqueda más adecuada para la investigación. Se aplicaron criterios de inclusión y exclusión, generando así varias

posibilidades. A continuación, en la tabla 2 mostraré la cadena de búsqueda que optimicé para obtener los siguientes resultados.

Tabla 2. Cadena de búsqueda, base de datos y resultados.

Cadena de búsqueda	Base de datos	Resultados
("CMMI" OR "Capability Maturity Model Integration") AND ("Software Quality" OR "Quality Management")		
("evaluation of software quality" OR "cmmi" OR "Capability Maturity Model Integration")	SCOPUS	113
("CMMI" OR "Capability Maturity Model Integration") AND ("Software Quality" OR "Quality Management")	IEEE	14
("CMMI" OR "Capability Maturity Model Integration") AND ("Software Quality")	ScienceDirect	37
("CMMI" OR "Capability Maturity Model Integration") AND ("Software Quality" OR "Quality Management")	ResearchGate	10

Análisis de resultados

Con el objetivo de refinar la información de la cadena de búsqueda, se obtuvieron los datos de los artículos encontrados en Scopus, IEEE Explore, Science Direct y ResearchGate donde se supo con la ayuda del Parsifal que, al momento de depurar las bases de datos indexadas, la cantidad obtenida para la revisión se redujo a 38 artículos relevantes.

Este resultado también se puede visualizar en la Fig. 3 que ilustra el número de artículos seleccionados de cada base de datos académica para realizar este estudio. Además, la Fig. 4 muestra el mecanismo para determinar la relevancia de cada estudio. Esto se hizo mediante la validación de los investigadores, de su citación, objetivos de claridad razonamiento de las propuestas teniendo en cuenta que también analizamos los años en los que se indexaron los

artículos. Esta información se puede observar en la Fig.5, donde refleja los filtros aplicados y las bases de datos que pasaron los debidos filtros.

Fig. 3 Artículos seleccionados de Scopus, IEEE Explore, Science Direct y ResearchGate.

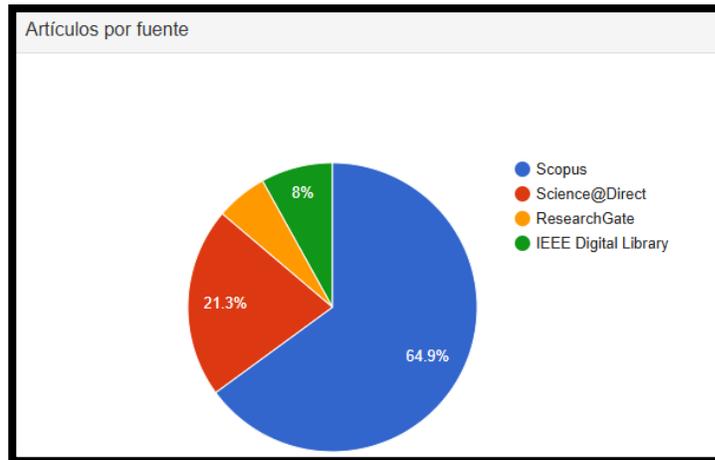


Fig. 4 Relevancia de la información a través de los años de publicación.

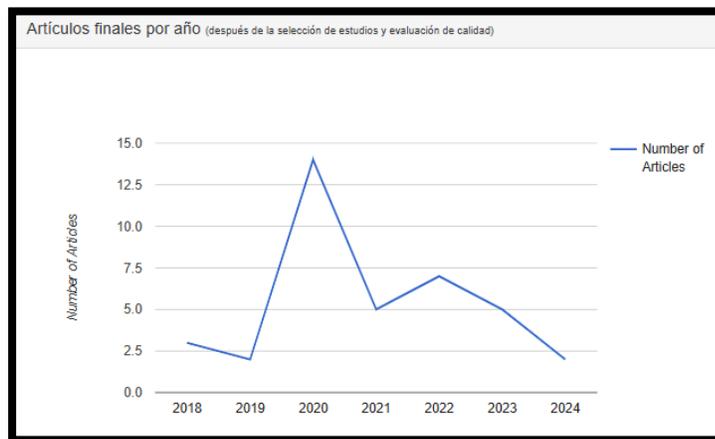
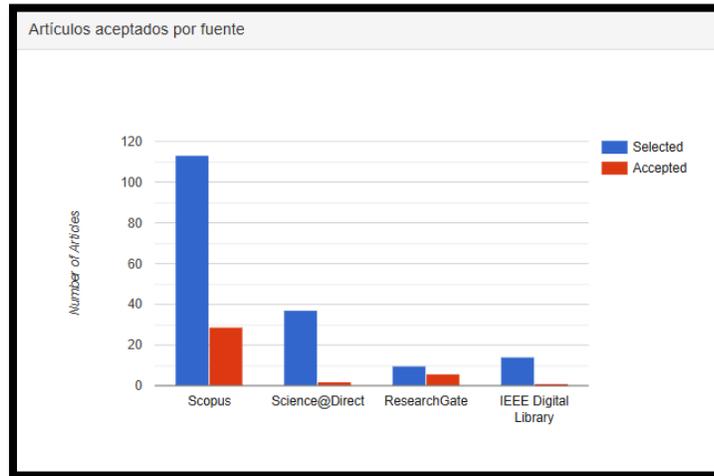


Fig. 5 Refleja las bases de datos que pasaron los debidos filtros para nuestra investigación.



Los documentos seleccionados, resultado del mapeo sistemático, se utilizaron para responder a las preguntas planteadas en la investigación. Estas preguntas, centrales en la investigación, se enfocan en la evaluación de la calidad del software en sistemas de educación superior utilizando el modelo CMMI.

En relación con la pregunta RQ1, se analizaron las principales metodologías y prácticas relacionadas con la implementación del modelo Capability Maturity Model Integration (CMMI) en centros de educación superior. Este análisis evidenció que el CMMI se utiliza como un marco estructurado para mejorar la calidad y la eficiencia de los procesos educativos y administrativos en estos entornos. Diversos estudios han adaptado el modelo a contextos académicos, integrando enfoques como el Educational CMMI, diseñado para evaluar proyectos educativos y alinear los objetivos pedagógicos con los principios de mejora continua del modelo (Amer et al., 2020; Tocto-Cano et al., 2020).

Por su parte, el trabajo de (Ramírez-Mora Sandra L et al., 2024) destacan cómo la madurez grupal, la eficiencia operativa y la efectividad del equipo son factores determinantes en la excelencia del desarrollo de software. Mediante un estudio de caso en una organización certificada en el Nivel 5 de CMMI-DEV, este trabajo ilustra cómo la implementación avanzada del

modelo no solo asegura la calidad técnica del software, sino también la consistencia organizacional. Este enfoque es particularmente relevante en el contexto educativo, donde la confiabilidad de los sistemas es esencial para garantizar el funcionamiento efectivo de los procesos académicos y administrativos.

La pregunta RQ2 explora las métricas más significativas en la implementación de CMMI en el contexto de la educación superior. Entre las métricas más destacadas se encuentran:

Indicadores clave de desempeño (KPIs): Estas métricas proporcionan una evaluación cuantitativa de la estabilidad y capacidad de los procesos, considerando tiempo, costo y alcance de los proyectos (Cerdeiral & Santos, 2019).

Cumplimiento de prácticas específicas: La gestión de requisitos y la validación del diseño son prácticas esenciales para garantizar el cumplimiento de estándares educativos establecidos (Amer et al., 2020; Humayun et al., 2022).

Satisfacción del usuario final: Esta métrica evalúa la experiencia de estudiantes y docentes, asegurando que el software cumpla no solo con requisitos técnicos, sino también con objetivos pedagógicos (Tocto-Cano et al., 2020).

Como se detalla en la Tabla 3, las métricas seleccionadas permiten evaluar distintos aspectos críticos de la calidad del software en sistemas de educación superior. Estas métricas fueron categorizadas en funcionalidad, confiabilidad, eficiencia, mantenibilidad, usabilidad, escalabilidad y seguridad, y se validaron mediante la comparación con estándares previamente establecidos en la literatura.

Tabla 3. Métricas relevantes para la evaluación de calidad de software según el modelo CMMI en centros educativos.

	Métrica	Descripción	Valor Actual	Meta	Ventajas
Funcionalidad	Tasa de cumplimiento de requisitos (%)	Porcentaje de requisitos funcionales implementados correctamente.	[Valor]	$\geq 90\%$	Asegura que el software cumple con los requerimientos específicos de estudiantes, docentes y personal, mejorando la satisfacción del usuario.
	Número de errores críticos	Número de fallos en funcionalidades principales durante pruebas o uso regular.	[Valor]	0	
Confiabilidad	Tasa de fallos por hora	Número de errores detectados por hora de uso del sistema.	[Valor]	≤ 0.1	Garantiza que el sistema sea estable y funcione sin interrupciones, reduciendo fallos en procesos críticos como inscripciones o calificaciones.
	Tasa de recuperación ante fallos (%)	Porcentaje de fallos solucionados exitosamente dentro de un tiempo definido.	[Valor]	$\geq 95\%$	
Eficiencia	Tiempo promedio de respuesta (segundos)	Tiempo medio que toma el sistema en responder a una solicitud del usuario	[Valor]	$\leq 2\text{ s}$	Optimiza el uso de recursos del sistema, asegurando tiempos de respuesta rápidos y una experiencia fluida en plataformas institucionales.
	Consumo de recursos (CPU, RAM)	Recursos consumidos por el sistema durante operaciones estándar (%).	[Valor]	$\leq 70\%$	
Mantenibilidad	Tiempo promedio para corregir errores (días)	Tiempo medio requerido para resolver fallos reportados.	[Valor]	$\leq 3\text{ días}$	Facilita actualizaciones y correcciones de errores de manera rápida y eficiente, reduciendo costos y tiempos de mantenimiento.
	Porcentaje de módulos probados	Porcentaje de componentes del sistema sometidos a pruebas exhaustivas.	[Valor]	$\geq 90\%$	
Usabilidad	Nivel de satisfacción del usuario (1-5)	Puntuación promedio en encuestas de satisfacción aplicadas a usuarios finales.	[Valor]	≥ 4.5	Mejora la facilidad de uso del sistema, permitiendo una experiencia intuitiva para estudiantes, docentes y personal administrativo.
	Tiempo promedio de aprendizaje (horas)	Tiempo requerido para que un nuevo usuario domine las funcionalidades básicas	[Valor]	$\leq 2\text{ h}$	
Escalabilidad	Capacidad de usuarios simultáneos	Número máximo de usuarios que el sistema puede soportar sin deterioro del rendimiento.	[Valor]	≥ 100	Permite que el software crezca y se adapte a un mayor número de usuarios o funciones sin pérdida

	Impacto de la adición de nuevas funciones	Tiempo promedio necesario para integrar y probar nuevas funcionalidades.	[Valor]	≤ 7 días	de rendimiento, apoyando el crecimiento institucional.
Seguridad	Número de incidentes de seguridad	Cantidad de violaciones de seguridad detectadas en el sistema.	[Valor]	0	Protege datos sensibles y asegura el cumplimiento de normativas de privacidad, garantizando la integridad y confidencialidad de la información.
	Tasa de éxito en pruebas de penetración (%)	Porcentaje de intentos de intrusión bloqueados exitosamente.	[Valor]	$\geq 95\%$	

Asimismo, el modelo Cloud Software Life Cycle Process (CSLCP), que combina prácticas del CMMI con características de computación en la nube, ha demostrado ser efectivo para mejorar sostenibilidad, escalabilidad y eficiencia en entornos educativos con recursos limitados (Alshazly et al., 2021; Xu et al., 2023). Por otro lado, enfoques híbridos como los propuestos por (Amer et al., 2020) integran métricas específicas de CMMI nivel 2 con metodologías ágiles como Scrum, introduciendo indicadores adicionales como porcentaje de defectos corregidos, cumplimiento de plazos y nivel de accesibilidad. Este enfoque resulta particularmente útil para evaluar la calidad del software en sistemas educativos dinámicos y adaptables. La pregunta RQ3 aborda cómo los principios y metodologías del modelo Capability Maturity Model Integration (CMMI) se adaptan y aplican de manera práctica en los sistemas educativos para mejorar la calidad y eficiencia de sus procesos. En este contexto, la implementación del modelo CMMI en instituciones de educación superior requiere de estrategias específicas que respondan a las particularidades del contexto educativo. Entre estas destacan:

Personalización de procesos: Los ciclos de vida de los proyectos se ajustan para alinearse con objetivos pedagógicos, priorizando áreas críticas como la recopilación de requisitos y la validación del diseño. Este enfoque asegura que tanto grandes universidades como instituciones con recursos limitados puedan beneficiarse del modelo (Schellekens et al., 2023).

Automatización tecnológica: Las herramientas digitales son esenciales para recolectar datos en tiempo real, permitiendo ajustes dinámicos y reduciendo esfuerzos manuales. Estas plataformas facilitan la escalabilidad de los sistemas educativos y promueven la optimización de recursos (Amer et al., 2020; Humayun et al., 2022).

Evaluaciones iterativas y colaborativas: La combinación de CMMI con metodologías ágiles, como Scrum, permite iteraciones regulares que aseguran flexibilidad y adaptación a cambios tecnológicos o pedagógicos. Además, la retroalimentación constante entre docentes, estudiantes y desarrolladores asegura que los sistemas cumplan con las necesidades específicas del entorno académico (Schellekens et al., 2023; Tocto-Cano et al., 2020).

Discusión

Este estudio ha demostrado cómo el modelo CMMI puede ser adaptado eficazmente a instituciones de educación superior para optimizar la calidad del software, especialmente en aspectos como funcionalidad, accesibilidad y sostenibilidad. Los hallazgos destacan que la personalización de prácticas y métricas específicas permite abordar desafíos únicos del sector educativo, como recursos limitados y una amplia diversidad de usuarios.

Los resultados del análisis indican que métricas como los KPIs, el cumplimiento de prácticas específicas y la satisfacción del usuario final son esenciales para evaluar y mejorar la calidad del software en sistemas educativos. Por ejemplo, el uso de indicadores de accesibilidad y tiempos de respuesta del sistema refleja cómo el modelo CMMI puede adaptarse para priorizar objetivos pedagógicos y operativos. Estos hallazgos no solo respaldan estudios previos (Amer et al., 2020; Tocto-Cano et al., 2020), sino que también amplían el alcance del modelo al incluir métricas centradas en usuarios finales, un aspecto poco explorado en la literatura previa.

En comparación con investigaciones como las de (Pries-Heje et al., 2023), que se enfocan en la detección temprana de problemas mediante síntomas, este estudio avanza al

proponer un marco integral que combina enfoques ágiles y tradicionales, maximizando la escalabilidad y la sostenibilidad del software educativo. Además, se refuerza la propuesta de (Humayun et al., 2022) sobre la importancia de la seguridad, añadiendo un enfoque específico en la accesibilidad y la usabilidad, dimensiones críticas para garantizar la inclusión en sistemas educativos. Por ejemplo, mientras que Six Sigma prioriza la reducción de defectos, el modelo CMMI puede integrar métricas más específicas de accesibilidad y usabilidad. Esto lo hace más adaptable para instituciones que buscan promover un aprendizaje inclusivo. (Khraiwesh, 2020)

Los hallazgos de esta investigación tienen implicaciones significativas tanto para los responsables de la gestión educativa como para desarrolladores de software:

Optimización de Procesos: La integración de herramientas automatizadas y enfoques iterativos permite a las instituciones educativas maximizar sus limitados recursos. Por ejemplo, el uso de plataformas de monitoreo en tiempo real facilita la toma de decisiones informadas, reduciendo errores y costos asociados a las operaciones manuales (Amer et al., 2020; Schellekens et al., 2023).

Mejoras continuas: La adopción de ciclos iterativos y retroalimentación constante fomenta la evolución de los sistemas de software, asegurando que permanezcan relevantes y efectivos frente a las demandas cambiantes de estudiantes y docentes (Schellekens et al., 2023; Tocto-Cano et al., 2020).

Adaptabilidad tecnológica: Este estudio valida que la combinación de CMMI con metodologías ágiles como Scrum es particularmente útil en entornos educativos dinámicos, donde los requisitos evolucionan constantemente. Esto es crucial para garantizar que los sistemas no solo sean funcionales, sino también alineados con objetivos pedagógicos (Humayun et al., 2022; Xu et al., 2023).

Aunque este trabajo presenta aportes significativos, existen limitaciones que deben considerarse. Por ejemplo:

Falta de validación empírica: Los hallazgos se basan en un análisis teórico, por lo que futuras investigaciones deberían incluir estudios de caso en instituciones educativas para corroborar la aplicabilidad de las métricas y enfoques propuestos.

Generalización restringida: Los resultados pueden variar según el tamaño, la ubicación y los recursos disponibles en las instituciones educativas, lo que subraya la necesidad de adaptar las prácticas del CMMI a contextos específicos.

Conclusión

Este estudio reafirma la relevancia del modelo CMMI como una herramienta versátil y eficaz para evaluar y mejorar la calidad del software en instituciones de educación superior. Los hallazgos subrayan que su implementación no solo permite garantizar estándares técnicos, sino también alinear los sistemas tecnológicos con objetivos pedagógicos clave, asegurando funcionalidad, accesibilidad y sostenibilidad.

La combinación de CMMI con metodologías ágiles como Scrum destaca como una estrategia integral para abordar las necesidades dinámicas del entorno educativo. Este enfoque optimiza la gestión de requisitos, la validación del diseño y la adaptabilidad tecnológica, lo que resulta crucial para mejorar la experiencia de estudiantes y docentes. Además, el uso de métricas específicas, como indicadores clave de desempeño (KPIs) y evaluaciones de satisfacción del usuario final, asegura que las soluciones tecnológicas no solo cumplan con estándares de calidad, sino que también se ajusten a las demandas de los usuarios.

Aunque el estudio proporciona una base teórica sólida, se identifican dos áreas clave para futuras investigaciones: la validación empírica y la generalización de los resultados. Se recomienda realizar estudios de caso en instituciones educativas reales para evaluar la

efectividad de las métricas propuestas, así como diseñar herramientas tecnológicas que faciliten la automatización de procesos y reduzcan los costos operativos. Estas iniciativas contribuirán a cerrar la brecha existente en la literatura, ofreciendo soluciones prácticas y efectivas que impulsen la transformación digital en el sector educativo.

Referencias bibliográficas

- Agrawal, M., & Chari, K. (2020). Impacts of process audit review and control efforts on software project outcomes. *IET Software*, 14(3), 293-299. <https://doi.org/10.1049/iet-sen.2019.0185>
- Aljedaibi, W., & Talal Agal, A. (2022). ASSESSMENT AND IMPLICATIONS OF APPLYING CMMI AT A SAUDI PUBLIC ORGANIZATION. *International Journal for Quality Research*, 16(4), 1197-1210. <https://doi.org/10.24874/IJQR16.04-15>
- Al-Matouq, H., Mahmood, S., Alshayeb, M., & Niazi, M. (2020). A Maturity Model for Secure Software Design: A Multivocal Study. *IEEE Access*, 8, 215758-215776. IEEE Access. <https://doi.org/10.1109/ACCESS.2020.3040220>
- Alshazly, A. A., ElNainay, M. Y., El-Zoghabi, A. A., & Abougabal, M. S. (2021). A cloud software life cycle process (CSLCP) model. *Ain Shams Engineering Journal*, 12(2), 1809-1822. <https://doi.org/10.1016/j.asej.2020.11.004>
- Amer, S. K., Badr, N., & Hamad, A. (2020). A SCRUM-BASED SOFTWARE QUALITY MODEL TO RAISE MATURITY OF SOFTWARE IN SME(S). . . Vol., 07.
- Astridita, A., Raharjo, T., & Fitriani, A. N. (2024). Perceived Benefits and Challenges of Implementing CMMI on Agile Project Management: A Systematic Literature Review. *International Journal of Advanced Computer Science and Applications*, 15(1). <https://doi.org/10.14569/IJACSA.2024.0150122>
- Ayyagari, M. R., & Atoum, I. (2019). CMMI-DEV Implementation Simplified. *International Journal of Advanced Computer Science and Applications*, 10(4). <https://doi.org/10.14569/IJACSA.2019.0100455>
- Barbara Kitchenham & Stuart M. Charters. (2007). (PDF) Guidelines for performing Systematic Literature Reviews in Software Engineering. *ResearchGate*.
-

https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering

Carvalho, J. V., Pereira, R. H., & Rocha, Á. (2020a). A systematic literature review on maturity models for information systems in higher education institutions. *Innovations in Education and Teaching International*, 57(4), 434-449. <https://doi.org/10.1080/14703297.2019.1648219>

Carvalho, J. V., Pereira, R. H., & Rocha, Á. (2020b). A systematic literature review on maturity models for information systems in higher education institutions. *Innovations in Education and Teaching International*, 57(4), 434-449. <https://doi.org/10.1080/14703297.2019.1648219>

Cerdeiral, C. T., & Santos, G. (2019). Software project management in high maturity: A systematic literature mapping. *Journal of Systems and Software*, 148, 56-87. <https://doi.org/10.1016/j.jss.2018.10.002>

Elfaki, A. O., & Bassfar, Z. (2020). Construction of a Software Development Model for Managing Final Year Projects in Information Technology Programmes. *International Journal of Emerging Technologies in Learning (iJET)*, 15(21), 4. <https://doi.org/10.3991/ijet.v15i21.15401>

Humayun, M., Jhanjhi, N., Fahhad Almufareh, M., & Ibrahim Khalil, M. (2022). Security Threat and Vulnerability Assessment and Measurement in Secure Software Development. *Computers, Materials & Continua*, 71(3), 5039-5059. <https://doi.org/10.32604/cmc.2022.019289>

Jadoon, G., Shafi, M., & Jan, S. (2020). Model Transformations Carried by the Traceability Framework for Enterprises in Software Industry. *The International Arab Journal of Information Technology*, 17(4A), 579-587. <https://doi.org/10.34028/iajit/17/4A/1>

- Jaiswal, K., & Garg, M. (2019). *Explore relationship between total quality management and CMMI in software development organisations: A study of Chandigarh and Mohali*. Vol.11 No.2, pp.154-170(2). <https://doi.org/10.1504/IJPOM.2019.100574>
- Keshta, I. (2022). A model for defining project lifecycle phases: Implementation of CMMI level 2 specific practice. *Journal of King Saud University - Computer and Information Sciences*, 34(2), 398-407. <https://doi.org/10.1016/j.jksuci.2019.10.013>
- Khraiwesh, M. (2020). Measures of Organizational Training in the Capability Maturity Model Integration (CMMI). *International Journal of Advanced Computer Science and Applications*, 11(2). <https://doi.org/10.14569/IJACSA.2020.0110274>
- Laitinen, K., Luhtala, M., Örmä, M., & Vaismaa, K. (2024). Productivity development enablers in the infrastructure sector: Capability maturity model integration approach. *Built Environment Project and Asset Management*, 14(2), 201-227. <https://doi.org/10.1108/BEPAM-07-2022-0095>
- Oyshi, F., Bonik, A., Shin, M. O. H., Rashid, J., Akter, M., Hasan, M., & Sadia, F. (2023). A Novel model to adapt CMMI Level 2 by Assessing the Local SMEs of Bangladesh. *Procedia Computer Science*, 219, 2043-2050. <https://doi.org/10.1016/j.procs.2023.01.506>
- Piowar-Sulej, K., Rojek-Nowosielska, M., Sokołowska-Durkalec, A., & Markowska-Przybyła, U. (2022). Maturity of CSR Implementation at the Organizational Level—From Literature Review to a Comprehensive Model. *Sustainability*, 14(24), 16492. <https://doi.org/10.3390/su142416492>
- Porto, D. D. P., Jesus, G. M. D., Ferrari, F. C., & Fabbri, S. C. P. F. (2021). Initiatives and challenges of using gamification in software engineering: A Systematic Mapping. *Journal of Systems and Software*, 173, 110870. <https://doi.org/10.1016/j.jss.2020.110870>
- Pries-Heje, J., Johansen, J., & Korsaa, M. (2023). Symptom-based improvement recommendations. *Journal of Software: Evolution and Process*, 35(8), e2375. <https://doi.org/10.1002/smr.2375>
-

- Ramírez-Mora Sandra L, Oktaba b Hanna, & Patlán Pérez Juana. (2024). Group maturity, team efficiency, and team effectiveness in software development: A case study in a CMMI-DEV Level 5 organization | Request PDF. *ResearchGate*. <https://doi.org/10.1002/smr.2232>
- Rojas Rodríguez, M., & Torres López, S. (2019). CMMI, un modelo de procesos que norma la calidad. *Serie Científica de la Universidad de las Ciencias Informáticas*, 12(10), 117-125.
- Schellekens, L. H., Van Der Schaaf, M. F., Van Der Vleuten, C. P. M., Prins, F. J., Wools, S., & Bok, H. G. J. (2023). Developing a digital application for quality assurance of assessment programmes in higher education. *Quality Assurance in Education*, 31(2), 346-366. <https://doi.org/10.1108/QAE-03-2022-0066>
- Tharun Thomas & P.G. Saleeshya. (2024). CMMI based fuzzy logic approach to assess the digital manufacturing maturity level of manufacturing industries | Request PDF. *ResearchGate*, Vol. 35(No. 8), 2658-2683. <https://doi.org/10.1108/TQM-07-2022-0235>
- Tocto-Cano, E., Paz Collado, S., López-Gonzales, J. L., & Turpo-Chaparro, J. E. (2020). A Systematic Review of the Application of Maturity Models in Universities. *Information*, 11(10), Article 10. <https://doi.org/10.3390/info11100466>
- Xu, M., Feng, C., Zhu, L., Zhu, S., & He, L. (2023). Grading evaluation of non-API OCTG quality level based on working conditions. *International Journal of Metrology and Quality Engineering*, 14, 8. <https://doi.org/10.1051/ijmqe/2023006>
- Yan, B., Yao, H.-P., Nakamura, M., Li, Z.-F., & Wang, D. (2020). A Case Study for Software Quality Evaluation Based on SCT Model With BP Neural Network. *IEEE Access*, 8, 56403-56414. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2020.2981872>
- Zarour, M., & Alenezi, M. (2021). Software process improvement initiative in medium size IT organization: A case study. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(1), 500. <https://doi.org/10.11591/ijeecs.v23.i1.pp500-509>
-