

<https://doi.org/10.69639/arandu.v13i1.2106>

Integración de herramientas digitales para el aprendizaje práctico de la ingeniería de software

Integration of Digital Tools for Practical Learning in Software Engineering

Maria Teodolinda Ortega Ovalle

maria.ortegao@up.ac.pa

<https://orcid.org/0009-0000-3629-9751>

Universidad de Panamá
Panamá

Artículo recibido: 18 febrero 2026-Aceptado para publicación: 20 marzo 2026
Conflictos de Interés: Ninguno que declarar

RESUMEN

La formación en Ingeniería de Software enfrenta el desafío constante de articular los fundamentos teóricos con la práctica profesional que demanda la industria tecnológica. En este contexto, la integración de herramientas digitales se ha convertido en un componente esencial para fortalecer el aprendizaje práctico, promover la colaboración y desarrollar competencias técnicas actualizadas. Este artículo analiza el papel pedagógico de plataformas como sistemas de control de versiones, entornos de desarrollo en la nube, herramientas de gestión ágil y tecnologías de automatización en la enseñanza universitaria de la Ingeniería de Software. A partir de una revisión teórica y una propuesta metodológica aplicada, se examina cómo estas herramientas facilitan la construcción de experiencias de aprendizaje auténticas mediante actividades basadas en proyectos, simulaciones de flujos de trabajo reales y prácticas de desarrollo colaborativo. Los resultados obtenidos evidencian mejoras significativas en la autonomía del estudiante, la calidad del código, la documentación técnica y la capacidad de trabajo en equipo. Asimismo, se observa un incremento en la motivación y en la apropiación de metodologías ágiles y buenas prácticas de ingeniería. Finalmente, se discuten las implicaciones pedagógicas de este enfoque, destacando la necesidad de una actualización continua del currículo y del rol docente como facilitador tecnológico. El artículo concluye con recomendaciones para la implementación institucional de estas herramientas en programas de Ingeniería de Software.


Palabras clave: Ingeniería de software; herramientas digitales; aprendizaje basado en proyectos; métricas de desempeño; Scrum

ABSTRACT

Software Engineering education continues to face the persistent challenge of bridging the gap between theoretical instruction and the practical demands of the technology industry. In this

context, the integration of digital tools has become a fundamental component for strengthening hands-on learning, fostering collaboration, and developing technical competencies aligned with current professional standards. This article examines the pedagogical role of platforms such as version control systems, cloud-based development environments, agile project management tools, and automation technologies within university-level Software Engineering programs. Through a theoretical review and an applied methodological proposal, the study explores how these tools support the creation of authentic learning experiences through project-based activities, simulations of real development workflows, and collaborative coding practices. The findings reveal significant improvements in student autonomy, code quality, technical documentation, and teamwork capabilities. Additionally, increased motivation and a stronger adoption of agile methodologies and engineering best practices were observed. The article discusses the pedagogical implications of this approach, emphasizing the need for continuous curriculum updates and the evolving role of the instructor as a technological facilitator. It concludes with recommendations for the institutional implementation of digital tools in Software Engineering education.

Keywords: Software engineering, digital tools, project-based learning, performance metrics, Scrum

Todo el contenido de la Revista Científica Internacional Arandu UTIC publicado en este sitio está disponible bajo licencia Creative Commons Attribution 4.0 International. 

INTRODUCCIÓN

La Ingeniería de Software es una disciplina dinámica, en constante transformación y profundamente vinculada a la práctica profesional. Su naturaleza exige que los estudiantes no solo comprendan los fundamentos teóricos, sino que también desarrollen habilidades técnicas, metodológicas y colaborativas que les permitan desenvolverse en entornos reales de desarrollo. Sin embargo, en muchos programas universitarios persiste una brecha significativa entre los contenidos impartidos en el aula y las herramientas, procesos y metodologías utilizadas en la industria tecnológica contemporánea. Esta desconexión limita la pertinencia de la formación y dificulta la transición de los estudiantes hacia escenarios laborales altamente demandantes.

En las últimas décadas, la integración de herramientas digitales ha revolucionado la forma en que se concibe, desarrolla, prueba y despliega software. Plataformas como GitHub, GitLab, Bitbucket, Visual Studio Code, Docker, Jira, Trello, Replit o GitHub Actions se han consolidado como componentes esenciales de los flujos de trabajo profesionales. Estas herramientas no solo facilitan la colaboración y la automatización, sino que también permiten implementar prácticas modernas como la integración continua, el despliegue continuo, la gestión ágil de proyectos y la virtualización de entornos de desarrollo. Incorporarlas en la educación superior representa una oportunidad estratégica para fortalecer el aprendizaje práctico y preparar a los futuros ingenieros de software con competencias alineadas a las exigencias del mercado.

Desde una perspectiva pedagógica, la integración de herramientas digitales se articula de manera natural con enfoques activos de enseñanza, tales como el aprendizaje basado en proyectos, el aprendizaje colaborativo y la educación centrada en competencias. Estos enfoques promueven la participación activa del estudiante, la resolución de problemas reales, la toma de decisiones en equipo y la reflexión sobre el propio proceso de aprendizaje. Las herramientas digitales potencian estas metodologías al permitir la simulación de escenarios profesionales, la documentación transparente del progreso, la retroalimentación inmediata y la colaboración distribuida. Como resultado, los estudiantes desarrollan no solo habilidades técnicas, sino también competencias transversales como comunicación efectiva, pensamiento crítico, adaptabilidad y responsabilidad profesional.

No obstante, la integración de estas herramientas en contextos educativos también plantea desafíos importantes. Las instituciones deben considerar aspectos como la infraestructura tecnológica, la capacitación docente, la actualización curricular y la alfabetización digital de los estudiantes. Además, la rápida evolución de las tecnologías requiere una revisión constante de los contenidos y prácticas pedagógicas para garantizar su vigencia y relevancia. Comprender cómo estas herramientas pueden incorporarse de manera efectiva y cómo influyen en el aprendizaje práctico es fundamental para modernizar la enseñanza de la Ingeniería de Software.

Este artículo tiene como propósito analizar el impacto pedagógico de la integración de herramientas digitales en el aprendizaje práctico de la Ingeniería de Software. Para ello, se presenta un marco teórico que contextualiza el uso de tecnologías educativas en esta disciplina, una metodología de implementación en entornos universitarios y un análisis de resultados derivados de experiencias aplicadas. Finalmente, se discuten las implicaciones de este enfoque y se ofrecen recomendaciones para su adopción institucional, con el fin de fortalecer la formación de ingenieros de software capaces de enfrentar los retos de una industria en constante evolución.

METODOLOGÍA

La metodología de este estudio se fundamenta en un enfoque mixto que combina elementos cualitativos y cuantitativos con el propósito de analizar de manera integral el impacto de la integración de herramientas digitales en el aprendizaje práctico de la Ingeniería de Software. Este enfoque permite comprender tanto los resultados observables en el desempeño de los estudiantes como las percepciones, experiencias y dinámicas que emergen durante el proceso formativo. El estudio adopta un diseño descriptivo-aplicado, orientado a caracterizar el uso de herramientas digitales en un entorno académico real y a evaluar su contribución al desarrollo de competencias técnicas y colaborativas. La intervención se implementó en un curso universitario de Ingeniería de Software en el que los estudiantes participaron en actividades prácticas basadas en proyectos y en simulaciones de flujos de trabajo profesionales.

La muestra estuvo conformada por estudiantes de tercer y cuarto año de la carrera, seleccionados mediante un muestreo no probabilístico e intencional, dado que formaban parte del grupo donde se desarrolló la intervención. La participación fue voluntaria y se garantizó la confidencialidad de la información recopilada. Para el desarrollo del estudio se utilizaron diversas herramientas digitales ampliamente empleadas en la industria del software, entre ellas GitHub o GitLab para el control de versiones, Visual Studio Code como entorno de desarrollo, Docker para la virtualización de entornos, Jira o Trello para la gestión ágil de tareas, GitHub Actions para la automatización y la integración continua, y plataformas como Replit o CodeSandbox para la programación en la nube. La selección de estas herramientas respondió a criterios de accesibilidad, pertinencia profesional, facilidad de adopción y potencial pedagógico.

El procedimiento se desarrolló en cuatro fases. En la primera se realizó un diagnóstico inicial para evaluar el nivel de familiaridad de los estudiantes con las herramientas digitales mediante un cuestionario y una actividad práctica. En la segunda fase se llevó a cabo un proceso de capacitación y acompañamiento que incluyó sesiones introductorias, guías prácticas y apoyo docente continuo. La tercera fase consistió en el desarrollo de proyectos en equipo siguiendo metodologías ágiles, durante la cual se registraron métricas reales de desempeño en las plataformas digitales, como la frecuencia de commits, la asignación y resolución de tareas y el uso de pipelines automatizados. Finalmente, en la cuarta fase se realizó la evaluación del proceso

mediante encuestas de percepción, análisis de repositorios y la aplicación de rúbricas centradas en la calidad técnica, la documentación y la colaboración.

El análisis de los datos se efectuó mediante estadística descriptiva para los resultados cuantitativos y análisis de contenido para la información cualitativa. La triangulación de ambas fuentes permitió validar los hallazgos y obtener una comprensión más profunda del impacto pedagógico de la integración de herramientas digitales en el aprendizaje práctico de la Ingeniería de Software.

Marco teórico

El aprendizaje práctico en la Ingeniería de Software ha sido ampliamente estudiado desde diversas perspectivas pedagógicas y tecnológicas. La literatura coincide en que la naturaleza de esta disciplina exige un equilibrio entre la comprensión teórica de los conceptos fundamentales y la aplicación directa de dichos conocimientos en contextos reales o simulados. Autores como Sommerville y Pressman destacan que la Ingeniería de Software se sustenta en procesos sistemáticos que requieren no solo dominio conceptual, sino también habilidades técnicas, metodológicas y colaborativas que se desarrollan mediante la práctica constante. En este sentido, la incorporación de herramientas digitales en entornos educativos se ha convertido en un elemento clave para fortalecer la formación profesional de los futuros ingenieros.

El uso de herramientas digitales en la educación superior se enmarca dentro de las teorías del aprendizaje activo, el constructivismo y el aprendizaje situado. Desde el enfoque constructivista, el estudiante construye conocimiento a partir de experiencias significativas, lo cual se potencia cuando interactúa con herramientas que replican escenarios reales de trabajo. Por su parte, el aprendizaje situado sostiene que las competencias se desarrollan de manera más efectiva cuando el estudiante participa en actividades auténticas que reflejan las prácticas de una comunidad profesional. En el caso de la Ingeniería de Software, estas prácticas incluyen el uso de sistemas de control de versiones, entornos de desarrollo integrados, plataformas de automatización y herramientas de gestión ágil, todas ellas ampliamente utilizadas en la industria tecnológica.

Diversos estudios han demostrado que la integración de herramientas como GitHub, GitLab, Docker, Jira o Visual Studio Code favorece el desarrollo de competencias técnicas esenciales. El control de versiones, por ejemplo, no solo permite gestionar el historial del código, sino que también fomenta la colaboración, la revisión entre pares y la responsabilidad compartida sobre el producto final. De manera similar, las herramientas de gestión ágil facilitan la organización del trabajo, la planificación iterativa y la comunicación dentro de los equipos, aspectos fundamentales en los entornos de desarrollo modernos. La automatización mediante pipelines de integración y despliegue continuo, por su parte, introduce al estudiante en prácticas avanzadas que mejoran la calidad del software y reducen errores humanos.

La literatura también resalta el papel del docente como facilitador tecnológico. La integración de herramientas digitales no implica únicamente su uso técnico, sino también la creación de experiencias de aprendizaje que permitan al estudiante comprender su propósito, aplicarlas de manera estratégica y reflexionar sobre su impacto en el proceso de desarrollo. El docente debe guiar, acompañar y contextualizar el uso de estas herramientas, promoviendo un aprendizaje autónomo y significativo. En este sentido, la capacitación docente y la actualización curricular son factores determinantes para garantizar una implementación efectiva.

Asimismo, la investigación en educación en ingeniería ha señalado que el aprendizaje basado en proyectos es uno de los enfoques más adecuados para integrar herramientas digitales. Este enfoque permite que los estudiantes trabajen en problemas reales o simulados, tomen decisiones técnicas, gestionen tareas, documenten procesos y colaboren de manera activa. Las herramientas digitales actúan como mediadoras del aprendizaje, proporcionando un entorno donde los estudiantes pueden experimentar, equivocarse, corregir y mejorar continuamente. Esto no solo fortalece las competencias técnicas, sino también habilidades transversales como la comunicación, el liderazgo, la resolución de problemas y la gestión del tiempo.

Finalmente, el marco teórico reconoce que la integración de herramientas digitales también presenta desafíos. Entre ellos se encuentran la brecha de alfabetización digital entre estudiantes, la resistencia al cambio por parte de algunos docentes, la necesidad de infraestructura tecnológica adecuada y la rápida obsolescencia de las herramientas. Sin embargo, la evidencia disponible indica que, cuando se implementan de manera planificada y pedagógicamente fundamentada, estas herramientas contribuyen significativamente a mejorar la calidad del aprendizaje práctico en Ingeniería de Software y a preparar a los estudiantes para enfrentar los retos de una industria en constante evolución.

RESULTADOS

Los resultados obtenidos a partir de la intervención pedagógica permiten analizar el impacto de la integración de herramientas digitales en el aprendizaje práctico de los estudiantes de Ingeniería de Software. El análisis se desarrolló a partir de tres fuentes principales: las métricas generadas en los repositorios de GitHub, las encuestas de percepción aplicadas al finalizar el curso y la evaluación de los proyectos finales mediante una rúbrica estandarizada. La triangulación de estos datos ofrece una visión amplia y consistente del desempeño estudiantil y de la efectividad del enfoque implementado.

En primer lugar, las métricas de desempeño en GitHub evidencian un uso activo y constante de las herramientas de control de versiones y colaboración. La Tabla 1 muestra que los equipos registraron un número significativo de commits, issues y pull requests, lo cual indica una participación sostenida en el desarrollo del proyecto. Los equipos C y A destacan por su mayor frecuencia de contribución semanal, lo que se refleja también en un mayor número de issues

resueltos y solicitudes de integración. Estos resultados sugieren que la adopción de herramientas digitales favoreció la organización del trabajo, la distribución de tareas y la colaboración continua entre los integrantes de cada equipo.

Tabla 1
Métricas de desempeño en GitHub por equipo

Equipo	Commits totales	Issues creados	Issues resueltos	Pull Requests	Frecuencia contribución
A	158	22	20	14	5.2
B	143	18	16	12	4.8
C	167	25	23	16	5.5
D	121	15	13	10	4.1
E	134	19	17	11	4.6

En segundo lugar, los resultados de la encuesta aplicada a los estudiantes reflejan una percepción altamente positiva respecto al uso de herramientas digitales en el proceso de aprendizaje. Como se observa en la Tabla 2, los promedios obtenidos en todos los ítems superan el valor de 4 en una escala de 1 a 5, lo que indica un alto nivel de satisfacción. Los estudiantes consideran que las herramientas facilitaron su aprendizaje práctico, mejoraron su capacidad de colaborar y les permitieron comprender mejor los procesos de integración y despliegue continuo. Asimismo, la valoración del acompañamiento docente fue especialmente alta, lo que resalta la importancia del rol del facilitador tecnológico en este tipo de intervenciones.

Tabla 2
Resultados de encuesta sobre percepción del aprendizaje práctico

Ítem evaluado	Promedio	Desviación estándar
Las herramientas digitales facilitaron mi aprendizaje práctico	4.6	0.48
Me siento más preparado para trabajar en proyectos reales	4.4	0.52
El uso de GitHub mejoró mi capacidad de colaborar	4.7	0.44
La automatización (CI/CD) mejoró mi comprensión del ciclo de desarrollo	4.3	0.57
El trabajo en equipo fue más organizado gracias a las herramientas	4.5	0.50
El docente brindó acompañamiento adecuado en el uso de herramientas	4.8	0.39

Un tercer conjunto de resultados corresponde a la comparación del nivel de dominio de herramientas digitales antes y después de la intervención. La Tabla 3 muestra incrementos

significativos en todas las competencias evaluadas, especialmente en el uso de control de versiones y en la comprensión de procesos de automatización e integración continua. Estas mejoras sugieren que la experiencia práctica, combinada con el uso sistemático de herramientas profesionales, contribuyó de manera directa al fortalecimiento de habilidades técnicas esenciales para la disciplina.

Tabla 3
Comparación antes/después del dominio de herramientas digitales

Competencia evaluada	Antes	Después	Variación
Uso de control de versiones	2.1	4.5	+2.4
Manejo de entornos de desarrollo	2.8	4.6	+1.8
Gestión ágil de tareas	2.4	4.3	+1.9
Automatización e integración continua	1.9	4.1	+2.2
Documentación técnica	2.6	4.2	+1.6
Trabajo colaborativo	3.0	4.7	+1.7

Finalmente, la evaluación de los proyectos finales confirma la tendencia observada en los datos anteriores. La Tabla 4 muestra que todos los equipos alcanzaron puntajes altos en calidad del código, documentación, uso de herramientas digitales y trabajo colaborativo. El equipo C obtuvo el puntaje más alto, destacándose por su excelente manejo de herramientas digitales y por la calidad técnica del producto final. Estos resultados evidencian que la integración de herramientas digitales no solo favoreció el proceso de aprendizaje, sino que también se reflejó en la calidad de los productos desarrollados.

Tabla 4
Evaluación final de proyectos (rúbrica)

Equipo	Calidad código	Documentación	Uso herramientas	Trabajo colaborativo	Puntaje total
A	9.2	8.7	9.5	9.0	36.4
B	8.8	8.1	9.0	8.6	34.5
C	9.4	9.0	9.7	9.3	37.4
D	8.2	7.9	8.5	8.1	32.7
E	8.6	8.3	8.9	8.4	34.2

En conjunto, los resultados permiten concluir que la integración de herramientas digitales tuvo un impacto positivo y significativo en el aprendizaje práctico de los estudiantes. Las métricas

objetivas, las percepciones estudiantiles y la calidad de los proyectos finales convergen en demostrar que el uso de estas herramientas favorece la colaboración, la autonomía, la organización del trabajo y el desarrollo de competencias técnicas esenciales para la Ingeniería de Software.

DISCUSIÓN

Los resultados obtenidos en este estudio permiten reflexionar sobre el impacto real que tiene la integración de herramientas digitales en el aprendizaje práctico de la Ingeniería de Software. En primer lugar, las métricas derivadas del uso de GitHub evidencian que los estudiantes no solo adoptaron las herramientas, sino que las utilizaron de manera constante y significativa. La frecuencia de commits, la resolución de issues y la generación de pull requests muestran que los equipos lograron reproducir dinámicas propias de entornos profesionales, lo cual coincide con lo planteado por la literatura sobre aprendizaje situado y aprendizaje basado en proyectos. El hecho de que los equipos con mayor constancia de trabajo presentaran también mejores indicadores de colaboración sugiere que el uso de estas herramientas favorece la organización del trabajo y la responsabilidad compartida.

En segundo lugar, las percepciones estudiantiles reflejan un alto nivel de satisfacción con el uso de herramientas digitales, especialmente en aspectos relacionados con la colaboración, la comprensión del ciclo de desarrollo y la preparación para escenarios reales. Estos hallazgos coinciden con estudios previos que señalan que las herramientas digitales no solo facilitan la gestión técnica del proyecto, sino que también fortalecen habilidades transversales como la comunicación, la coordinación y la toma de decisiones. La valoración positiva del acompañamiento docente confirma la importancia del rol del facilitador tecnológico, quien no solo introduce las herramientas, sino que guía su uso pedagógico y contextualizado.

La comparación antes y después del dominio de competencias digitales muestra mejoras significativas en todas las áreas evaluadas. Este incremento evidencia que la experiencia práctica, mediada por herramientas profesionales, permite a los estudiantes desarrollar habilidades que difícilmente podrían adquirirse mediante métodos tradicionales basados exclusivamente en teoría o ejercicios aislados. El aumento notable en competencias como control de versiones y automatización sugiere que los estudiantes lograron apropiarse de prácticas avanzadas que son esenciales en la industria actual, lo cual refuerza la pertinencia de integrar estas herramientas en el currículo.

La evaluación final de los proyectos confirma que el uso de herramientas digitales no solo impacta el proceso de aprendizaje, sino también la calidad del producto final. Los equipos que demostraron mayor dominio de las herramientas obtuvieron mejores resultados en calidad del código, documentación y trabajo colaborativo. Esto sugiere que la integración de herramientas digitales no debe entenderse únicamente como un complemento tecnológico, sino como un

componente pedagógico que transforma la manera en que los estudiantes conciben, desarrollan y evalúan software.

Sin embargo, los resultados también permiten identificar desafíos importantes. Aunque la mayoría de los estudiantes mostró avances significativos, algunos equipos presentaron menor frecuencia de contribución y menor dominio de ciertas herramientas, lo cual podría estar relacionado con brechas de alfabetización digital, diferencias en el ritmo de aprendizaje o dificultades iniciales en la adopción de nuevas tecnologías. Esto refuerza la necesidad de ofrecer acompañamiento continuo, materiales de apoyo y estrategias diferenciadas que permitan atender las necesidades de todos los estudiantes.

Otro aspecto relevante es la necesidad de actualización constante. Las herramientas digitales evolucionan rápidamente, lo que implica que los programas académicos deben revisar periódicamente sus contenidos y metodologías para garantizar su vigencia. Asimismo, la capacitación docente se vuelve un elemento crítico, ya que el éxito de estas iniciativas depende en gran medida de la capacidad del docente para integrar las herramientas de manera pedagógicamente significativa.

En conjunto, la discusión de los resultados permite concluir que la integración de herramientas digitales en la enseñanza de la Ingeniería de Software no solo es pertinente, sino necesaria para preparar a los estudiantes frente a las demandas de la industria. Las herramientas digitales actúan como mediadoras del aprendizaje, fortalecen la práctica profesional, promueven la colaboración y permiten desarrollar competencias técnicas y transversales esenciales. No obstante, su implementación requiere planificación, acompañamiento docente y una visión institucional que reconozca su valor estratégico en la formación de ingenieros de software.

CONCLUSIÓN

Los resultados de este estudio permiten afirmar que la integración de herramientas digitales en el aprendizaje práctico de la Ingeniería de Software constituye una estrategia pedagógica altamente efectiva y pertinente para la formación de futuros profesionales en el área. La evidencia recopilada demuestra que el uso sistemático de plataformas como GitHub, Visual Studio Code, Docker, Jira y herramientas de automatización no solo facilita la comprensión de los procesos de desarrollo de software, sino que también promueve la colaboración, la organización del trabajo y la adquisición de competencias técnicas avanzadas que son esenciales en la industria tecnológica contemporánea.

El análisis de las métricas de desempeño, las percepciones estudiantiles y la calidad de los proyectos finales revela que los estudiantes lograron apropiarse de prácticas profesionales que difícilmente podrían desarrollarse mediante métodos tradicionales. La mejora significativa en competencias como el control de versiones, la gestión ágil, la automatización y la documentación técnica confirma que las herramientas digitales actúan como mediadoras del aprendizaje,

permitiendo experiencias auténticas que replican escenarios reales de trabajo. Asimismo, la alta valoración del acompañamiento docente subraya la importancia del rol del facilitador tecnológico en la orientación, contextualización y uso pedagógico de estas herramientas.

No obstante, la implementación de estas tecnologías también plantea desafíos que deben ser considerados. La necesidad de capacitación docente, la actualización curricular constante y la atención a las brechas de alfabetización digital son elementos clave para garantizar una adopción efectiva y sostenible. A pesar de estas limitaciones, los hallazgos del estudio indican que los beneficios superan ampliamente las dificultades, y que la integración de herramientas digitales representa un paso necesario para modernizar la enseñanza de la Ingeniería de Software y responder a las demandas de una industria en permanente evolución.

En síntesis, la incorporación de herramientas digitales no debe entenderse como un complemento opcional, sino como un componente esencial del currículo. Su integración fortalece el aprendizaje práctico, potencia el desarrollo de competencias profesionales y prepara a los estudiantes para enfrentar con éxito los retos del entorno laboral. Este estudio aporta evidencia concreta sobre su impacto positivo y ofrece una base sólida para que instituciones educativas, docentes y diseñadores curriculares avancen hacia modelos formativos más actualizados, pertinentes y alineados con las necesidades reales del sector tecnológico.

Recomendaciones

A partir de los resultados obtenidos y del análisis realizado, se proponen una serie de recomendaciones orientadas a fortalecer la integración de herramientas digitales en la enseñanza de la Ingeniería de Software. En primer lugar, se sugiere que las instituciones educativas incorporen de manera formal estas herramientas dentro del currículo, no como actividades complementarias, sino como componentes estructurales de los cursos prácticos. La experiencia demuestra que su uso sistemático favorece el desarrollo de competencias técnicas y colaborativas esenciales, por lo que su inclusión debe planificarse desde el diseño curricular y no depender únicamente de iniciativas aisladas del docente.

Asimismo, se recomienda establecer programas de capacitación continua para el profesorado, de modo que los docentes puedan actualizarse en el uso pedagógico y técnico de las herramientas digitales. La evidencia indica que el acompañamiento docente es un factor determinante para el éxito de estas experiencias, por lo que resulta indispensable que los profesores cuenten con el conocimiento, la confianza y los recursos necesarios para guiar a los estudiantes en su uso. Esta capacitación debe incluir no solo aspectos técnicos, sino también estrategias didácticas que permitan integrar las herramientas dentro de metodologías activas como el aprendizaje basado en proyectos.

Otra recomendación importante es promover la creación de entornos de práctica que simulen escenarios reales de desarrollo de software. Esto implica no solo utilizar herramientas digitales, sino también diseñar proyectos auténticos, fomentar la colaboración entre equipos y

establecer dinámicas de trabajo que reflejen las prácticas de la industria. La simulación de entornos profesionales contribuye a que los estudiantes comprendan el propósito de las herramientas y desarrollen competencias transferibles al ámbito laboral.

También se sugiere implementar mecanismos de seguimiento y retroalimentación continua durante el desarrollo de los proyectos. El análisis de métricas generadas por plataformas como GitHub ofrece información valiosa sobre el desempeño de los estudiantes, por lo que su uso debe integrarse como parte del proceso de evaluación formativa. La retroalimentación basada en datos permite identificar dificultades, orientar el trabajo colaborativo y mejorar la calidad del producto final.

Finalmente, se recomienda que las instituciones fortalezcan la infraestructura tecnológica necesaria para garantizar el acceso equitativo a las herramientas digitales. Esto incluye asegurar conectividad adecuada, disponibilidad de equipos, acceso a plataformas en la nube y soporte técnico oportuno. La reducción de brechas tecnológicas es fundamental para que todos los estudiantes puedan participar plenamente en las actividades prácticas y aprovechar los beneficios de las herramientas digitales.

En conjunto, estas recomendaciones buscan consolidar un modelo educativo más actualizado, pertinente y alineado con las demandas de la industria del software. La integración efectiva de herramientas digitales no solo mejora el aprendizaje práctico, sino que contribuye a formar profesionales más competentes, autónomos y preparados para enfrentar los retos de un entorno tecnológico en constante evolución.

Limitaciones del estudio

Aunque los resultados obtenidos en este estudio evidencian el impacto positivo de la integración de herramientas digitales en el aprendizaje práctico de la Ingeniería de Software, es importante reconocer una serie de limitaciones que condicionan el alcance y la generalización de los hallazgos. En primer lugar, el estudio se desarrolló con una muestra relativamente pequeña y perteneciente a un único curso universitario, lo cual restringe la posibilidad de extrapolar los resultados a otros contextos educativos o a poblaciones más amplias. La naturaleza intencional del muestreo también implica que los participantes ya tenían cierto interés o predisposición hacia el uso de tecnologías, lo que podría haber influido en su nivel de participación y en las percepciones reportadas.

Otra limitación relevante se relaciona con la variabilidad en el nivel de alfabetización digital de los estudiantes. Aunque la mayoría logró adaptarse al uso de las herramientas, algunos presentaron dificultades iniciales que pudieron afectar su desempeño y su experiencia de aprendizaje. Estas diferencias individuales no fueron controladas de manera exhaustiva, lo que introduce un margen de variabilidad que debe considerarse al interpretar los resultados. Asimismo, el estudio dependió en gran medida de la autopercepción de los estudiantes, lo cual,

si bien aporta información valiosa, puede estar sujeto a sesgos de deseabilidad social o a interpretaciones subjetivas de su propio progreso.

El análisis de métricas provenientes de plataformas como GitHub también presenta limitaciones. Si bien estas métricas ofrecen datos objetivos sobre la actividad de los estudiantes, no siempre reflejan la calidad del aprendizaje o la profundidad conceptual alcanzada. Por ejemplo, un mayor número de commits no necesariamente implica un mejor entendimiento del proceso de desarrollo, sino que puede responder a estrategias de trabajo fragmentado o a dinámicas internas de cada equipo. Además, las métricas no capturan aspectos cualitativos del proceso, como la calidad de las discusiones técnicas o la toma de decisiones colaborativas.

Finalmente, el estudio se desarrolló en un entorno controlado y con acompañamiento docente constante, lo cual pudo haber influido positivamente en la adopción de las herramientas. En contextos con menor apoyo institucional o con docentes menos familiarizados con estas tecnologías, los resultados podrían diferir. La rápida evolución de las herramientas digitales también representa un desafío, ya que los hallazgos pueden perder vigencia si no se actualizan periódicamente las prácticas y los recursos utilizados.

A pesar de estas limitaciones, el estudio ofrece evidencia sólida sobre los beneficios de integrar herramientas digitales en la enseñanza de la Ingeniería de Software y sienta las bases para futuras investigaciones que profundicen en su impacto, amplíen la muestra y exploren nuevas variables relacionadas con la práctica profesional y el aprendizaje autónomo.

REFERENCIAS

- Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley.
- Beck, K. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Addison-Wesley.
- Boettger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1), 71–79. <https://doi.org/10.1145/2723872.2723882>
- Hazzan, O., & Dubinsky, Y. (2014). *Agile software engineering*. Springer. <https://doi.org/10.1007/978-3-319-05032-4>
- Hilton, M., Tunnell, T., Huang, K., Marinov, D., & Dig, D. (2016). Usage, costs, and benefits of continuous integration in open-source projects. *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 426–437. <https://doi.org/10.1145/2970276.2970358>
- Kolb, D. A. (2015). *Experiential learning: Experience as the source of learning and development* (2nd ed.). Pearson.
- Paasivaara, M., Heikkilä, V. T., Lassenius, C., & Toivola, T. (2014). Teaching students Scrum using LEGO blocks. *Proceedings of the 35th International Conference on Software Engineering (ICSE)*, 1–10. <https://doi.org/10.1145/2591062.2591169> (doi.org in Bing)
- Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Prince, M. J., & Felder, R. M. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, 95(2), 123–138. <https://doi.org/10.1002/j.2168-9830.2006.tb00884.x>
- Rodríguez, P., Markkula, J., Oivo, M., & Turula, K. (2015). Survey on agile and lean usage in Finnish software industry. *Empirical Software Engineering*, 20(5), 1384–1425.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum guide*. Scrum.org. <https://scrumguides.org>
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- Spinellis, D. (2012). Git. *IEEE Software*, 29(3), 100–101. <https://doi.org/10.1109/MS.2012.61>
- Thomas, J. W. (2000). *A review of research on project-based learning*. Autodesk Foundation.