TECNOLOGÍA: RESULTADOS DE INVESTIGACIÓN

DOI: https://doi.org/10.35588/djv1kq31

Eficacia en el registro de documentación en equipos de trabajo con desarrollo de software

Effectiveness in documentation recording in software development work environments

Eficácia no registro de documentação em equipes de desenvolvimento de software

Edición Nº55 – Abril de 2026

Artículo Recibido: 22 de agosto de 2025

Aprobado: 15 de mayo de 2026

Publicado: 19 de mayo de 2026

Autoras:

Yuliana Sánchez-Acosta[1] y María-de-Jesús Hernández-Garza[2]

Resumen:

El presente artículo de investigación aborda la eficacia de la documentación como un factor crítico, pero frecuentemente subestimado para la sostenibilidad de los proyectos de software. Partiendo de la problemática de que muchos proyectos son abandonados por una gestión deficiente y una creciente complejidad técnica, este estudio se propuso

[1] Mst. Universidad Autónoma de Nuevo León. Monterrey, México. Correo electrónico: yuliana.sanchezacst@uanl.edu.mx, https://orcid.org/0009-0004-6876-0513
[2] Dra. Universidad Autónoma de Nuevo León. Monterrey, México. Correo electrónico: maria.hernandezgza@uanl.edu.mx, https://orcid.org/0009-0004-4031-1986

cuantificar objetivamente las prácticas de documentación y versionamiento en un equipo de desarrollo de una institución financiera. La investigación se fundamenta en un marco teórico que conecta la claridad del código, las metodologías ágiles y el control de versiones con la productividad del equipo y la viabilidad del proyecto a largo plazo. La metodología empleada consistió en un análisis cuantitativo y empírico, utilizando un script automatizado para examinar 1,294 commits de cuatro repositorios de software críticos durante el segundo trimestre de 2025. El instrumento midió la adherencia al estándar de *Conventional Commits* como un indicador de la calidad y disciplina en la documentación de cambios. Los resultados revelaron una baja tasa de cumplimiento consolidada de solo 24.34%, evidenciando una brecha significativa entre las buenas prácticas recomendadas y su aplicación real.

Palabras clave: *Conventional Commits*, documentación, versionamiento.

Abstract:

This research article addresses the effectiveness of documentation as a critical but frequently underestimated factor for the sustainability of software projects. Starting from the premise that many projects are abandoned due to deficient management and increasing technical complexity, this study aimed to objectively quantify documentation and versioning practices within a work team engaged in software development at a financial institution. The research is grounded in a theoretical framework that connects code clarity, agile methodologies, and version control with team productivity and long-term project viability. The methodology consisted of a quantitative and empirical analysis, using an automated script to examine 1,294 commits from four critical software repositories during the second quarter of 2025. The instrument measured adherence to the *Conventional Commits* standard as an indicator of quality and discipline in change documentation. The results revealed a low consolidated compliance rate of only 24.34%, evidencing a significant gap between recommended best practices and their actual application.

Keywords: Conventional Commits, documentation, versioning.

Resumo:

O presente artigo de pesquisa aborda a eficácia da documentação como um fator crítico, mas frequentemente subestimado, para a sustentabilidade de projetos de software. Partindo da problemática de que muitos projetos são abandonados devido a uma gestão deficiente e à crescente complexidade técnica, este estudo propôs-se a quantificar objetivamente as práticas de documentação e versionamento em uma equipe de trabalho com desenvolvimento de software de uma instituição financeira. A pesquisa fundamenta-se em um arcabouço teórico que conecta a clareza do código, as metodologias ágeis e o controle de versões com a produtividade da equipe e a viabilidade do projeto a longo prazo. A metodologia empregada consistiu em uma análise quantitativa e empírica, utilizando um script automatizado para examinar 1.294 commits de quatro repositórios de software críticos durante o segundo trimestre de 2025. O instrumento mediu a aderência ao padrão *Conventional Commits* como um indicador da qualidade e disciplina na documentação de mudanças. Os resultados revelaram uma baixa taxa de conformidade consolidada de apenas 24,34%, evidenciando uma lacuna significativa entre as boas práticas recomendadas e sua aplicação real.

Palavras-chave: *Conventional Commits*, documentação, versionamento.

1. INTRODUCCIÓN

El desarrollo de software tiene sus orígenes en la década de 1940, cuando surgieron las primeras computadoras de propósito general como ENIAC en Estados Unidos. En esta etapa inicial, el software no se consideraba una entidad separada del hardware, y la programación se realizaba mediante conexiones físicas y lenguaje máquina.

Posteriormente, entre 1950 y 1960, se originó el concepto formal de software a medida que las empresas comenzaron a implementar computadoras para el procesamiento de datos comerciales. Fue en esta época cuando nacieron lenguajes de alto nivel como FORTRAN y COBOL, facilitando la abstracción de los procesos; al respecto, Qin, Zhang, Qin, Xu, Dimitrov, Wang y Yu (2018) señalan que la creciente versatilidad computacional

impulsó estas exigencias de software, eliminando las barreras entre el hardware y el lenguaje de programación.

Durante las décadas de 1960 y 1980, el mundo enfrentó la denominada "crisis del software", caracterizada por proyectos que excedían presupuestos, incumplían plazos y presentaban baja calidad debido a la creciente complejidad de los sistemas. En respuesta a esto, en 1968, durante una conferencia patrocinada por la OTAN en Garmisch, Alemania, se acuñó el término "ingeniería de software".

A partir de los años 90 y con la llegada de Internet, el desarrollo de software se transformó radicalmente. Como explican Hoda, Norsaremah y Grundy (2018), aunque la crisis del software evidenció la creciente complejidad de la programación, la evolución posterior permitió que metodologías ágiles como Scrum se expandieran más allá del desarrollo tradicional, integrándose en implementaciones modernas como DevOps para una entrega de valor continua.

En la actualidad, la importancia del desarrollo de software radica en su papel fundamental para la digitalización y optimización de los procesos empresariales. La automatización de tareas, la gestión de la información y la mejora en la toma de decisiones dependen intrínsecamente de sistemas de software robustos. Haciendo énfasis en la opinión de Ávila-Guerrero, Bernal-Díaz y Monroy-Gómez (2023), la transformación digital corporativa implica adaptar y reconvertir tecnologías que ya forman parte de las rutinas diarias de las empresas para incrementar su eficiencia.

Tecnologías digitales y optimización de recursos son elementos clave en la modernización empresarial, por lo que Aponte (2024) destaca que su integración en las operaciones busca fundamentalmente mejorar la eficiencia de los procesos y facilitar la toma de decisiones reduciendo la necesidad de intervención humana. En el contexto del desarrollo de software, esta eficiencia se traduce en la necesidad de herramientas automatizadas de versionamiento que minimicen los errores manuales y estandaricen el registro de cambios en el código.

A menudo existe una carencia en el mantenimiento regular de los proyectos, y de acuerdo con Ait, Cánovas-Izquierdo y Cabot (2022), muchos son abandonados poco después de su creación. La raíz de este problema la mayoría de las veces reside en que el software complejo y la alta calidad son, para Gracia-Orejuela (2022), características de los sistemas actuales, que se vuelven más exigentes.

Los entornos de desarrollo colaborativo tienen la posibilidad de aumentar las condiciones de calidad del código y facilitar la participación de los desarrolladores desde cualquier sitio, por medio de los avances en herramientas de versionamiento y documentación técnica, según Nurlanuly, Araz y Zhuldyz (2025).

El teletrabajo era una opción que no se planeaba hace tiempo atrás y en la actualidad ya existe sin tomarle la importancia necesaria. De acuerdo con la experiencia de Espinoza-Calderón (2025), esta creciente complejidad, si no se gestiona adecuadamente, tiene repercusiones económicas directas; el equipo de calidad y el valor económico se ven afectados por la ausencia del análisis y el seguimiento a los empleados que laboran desde casa, causando graves consecuencias.

Tal como mencionan Rozo-Rodríguez, Casanovas y Pollo-Cattaneo (2020), la calidad del producto del software es afectado por la cantidad extensa de volumen de conocimiento obtenido por los procesos y las actividades al administrar la calidad de la ingeniería del software.

El producto de software y la tecnología GIT permiten que la escritura de código sea fácilmente manejable, según lo expuesto por Chakraborty y Aithal (2022). La función principal de esta tecnología es que el control de versiones, en base a lo que menciona Slama, Ismail y Latrach (2023), ayuda a los desarrolladores a rastrear y gestionar los cambios en el código de un proyecto.

Complementariamente, metodologías como Scrum organizan el flujo de trabajo, donde el Product Owner y las necesidades del cliente son transmitidas por él al equipo de desarrollo, como explican Pachacama-Granda, Morales-Cazar, Vera-Toscano y Pallo-Morales (2024).

Herramientas de inteligencia artificial han demostrado ser un gran apoyo, donde los programadores sin experiencia y el beneficio de IA es mayor en este grupo, como sugieren los resultados del estudio de Peng, Kalliamvakou, Cihon y Demirer (2023).

El alcance de este estudio se delimita al análisis cuantitativo de la adherencia al estándar *Conventional Commits*. Como limitación principal, la investigación se restringió a una única institución financiera, evaluando exclusivamente el segundo trimestre de 2025 y analizando cuatro repositorios específicos pertenecientes al flujo de "Onboarding Digital", lo que sugiere cautela al generalizar los resultados a otras industrias.

El objetivo principal de este estudio es evaluar el nivel de cumplimiento del estándar *Conventional Commits* en los repositorios de código de una institución financiera. Para ello, se plantea la siguiente pregunta de investigación: ¿En qué medida los desarrolladores de la institución se adhieren a las prácticas de documentación estructurada en el versionamiento de software?

1. ANTECEDENTES TEÓRICOS

2.1. Fundamentos del desarrollo de software

El desarrollo de software en la era digital se ha convertido en una disciplina compleja que requiere una gestión meticulosa para evitar el fracaso prematuro de los proyectos. La sostenibilidad de un proyecto a menudo depende de prácticas robustas que van más allá de la simple escritura de código.

El desarrollo de software y cooperación efectiva dependen directamente de la calidad de los registros técnicos, motivo por el cual Nurlanuly, Araz y Zhuldyz (2025) proponen integrar buenas prácticas de documentación desde las etapas iniciales del ciclo de vida del proyecto, abarcando aspectos técnicos y administrativos, lo que prepara a los equipos para los desafíos de mantenibilidad del sector.

Documentación técnica y seguimiento de proyectos son pilares fundamentales para la eficiencia de los equipos, y según Álvarez-Fuentes (2025), las herramientas tecnológicas apoyan la transformación de las organizaciones al centralizar el conocimiento técnico. Esta formación en el uso de estándares es crucial, ya que mejora la calidad del producto final y crea una cultura de excelencia desde las etapas tempranas del desarrollo.

La incorporación de estándares de documentación es una estrategia esencial para aplicarla en entornos de desarrollo colaborativo con el fin de asegurar la trazabilidad del código y la continuidad del conocimiento técnico, tal como lo evidencia el trabajo de Álvarez-Fuentes (2025).

Analizando el punto de visa de Leones-Zambrano, Macias-Bazurto, Pilla-Zuniga y Fernández-Sánchez (2024), un sistema de versionamiento claro es vital para asegurar la compatibilidad de nuevas versiones con clientes existentes, evitando problemas de integración y asegurando la continuidad del servicio para los usuarios finales.

2.2. Documentación y trazabilidad

Equipos de desarrollo y operaciones requieren puentes de entendimiento sólidos, y como comenta Acosta-Bravo (2019), las prácticas DevOps han transformado la industria al promover espacios colaborativos mediante herramientas integradas de control de versiones. Este modelo colaborativo es directamente aplicable al mantenimiento de la documentación de software, donde múltiples miembros del equipo contribuyen simultáneamente al mismo repositorio.

En base a lo que comenta Aponte (2024), las tecnologías digitales aplicadas a la economía circular contribuyen a la disminución de residuos y facilitan la reutilización de recursos; análogamente, en el ecosistema del software, la documentación técnica y el versionamiento actúan como mecanismos de reutilización de conocimiento, previniendo el desperdicio de esfuerzo técnico y reduciendo la redundancia operativa en los equipos de desarrollo.

2.3. Gestión empresarial

La elección de una metodología adecuada puede ser la diferencia entre el éxito y el fracaso. La colaboración entre diferentes áreas es un pilar de las metodologías modernas. DevOps y la cooperación estrecha entre desarrollo y operaciones definen este enfoque, según la investigación de Gracia-Orejuela (2022).

2.4. Equipos ágiles y productividad

La productividad está influenciada por la comunicación interna, el compromiso y la confianza entre los miembros del equipo. Guerrero-Calvache y Hernández (2023) explican que estos elementos son esenciales para medir el desempeño en entornos ágiles, junto con la velocidad operativa y la satisfacción del cliente. Esta comunicación interna se materializa, en gran medida, a través de los mensajes de commit, los cuales actúan como el canal principal de diálogo asíncrono entre los desarrolladores de un equipo distribuido.

Dada la rápida evolución de las tecnologías, Pastrana-Pardo, Ordoñez-Erazo y Cobos-Lozada (2022) señalan que es imperativo para la sociedad incrementar la adaptabilidad de sus procesos de producción, potenciando su agilidad y efectividad. Para lograr dicha adaptabilidad en el desarrollo de software, los equipos dependen de repositorios bien estructurados que faciliten la asimilación rápida de nuevos requerimientos y la comprensión del historial de cambios.

Castillo-Rodríguez (2023), sostiene que la flexibilidad y el teletrabajo son requisitos fundamentales para las nuevas generaciones al buscar empleo en diversas organizaciones. Sin embargo, el éxito del teletrabajo en ingeniería de software requiere de una documentación estandarizada que evite la dependencia de la comunicación presencial para comprender el estado del código fuente.

2.5. Calidad en la documentación del desarrollo de software

Una convención ligera y un conjunto sencillo de reglas es lo que ofrece esta especificación, la cual, según *Conventional Commits* (2023), se integra con el versionamiento semántico al clasificar los cambios en categorías como feat (nuevas características), fix (corrección de errores), docs (documentación), refactor (reestructuración de código), style (cambios de formato), test (pruebas) y chore (tareas de mantenimiento), entre otros, permitiendo una trazabilidad semántica del historial del proyecto.

El estándar *Conventional Commits* es una especificación ligera que se superpone a los mensajes de commit para crear un historial explícito y legible tanto por humanos como por máquinas.

Es importante mencionar que, un rol importante y comunicación efectiva son fundamentales en el desarrollo de software colaborativo, motivo por el cual Li y Ahmed (2023) afirman que un mensaje de commit de calidad bajo este estándar debe incluir tanto el resumen de la modificación técnica (qué) como la motivación subyacente (por qué), ya que la calidad de los mensajes impacta directamente en la propensión a defectos del software.

Un rol crucial y una descripción clara son características que los mensajes de commit aportan al desarrollo colaborativo, aunque Ma y Lopes (2023) advierten que la ausencia de información útil y estandarizada en estos textos puede perjudicar severamente la comunicación entre desarrolladores y el mantenimiento futuro del código fuente. Fue

precisamente este estándar el criterio de comparación utilizado en el presente estudio para evaluar la calidad de los commits registrados en los repositorios analizados.

En este contexto, la calidad de la documentación es un factor clave. Como indica Espinoza-Calderón (2025), la falta de versionamiento provoca que los miembros trabajen en documentos desactualizados, generando redundancias y perdiendo tiempo valioso.

Por lo mencionado en líneas anteriores, se considera esencial el análisis de la forma en que los empleados ejecutan el desarrollo de software, y comparten simultáneamente los proyectos para su modificación en base a los requerimientos de la organización y la eficacia requerida al momento de hacer el registro de las líneas de código cambiadas usando herramientas de versionamiento para una mejor calidad de documentación.

Hipótesis planteadas

H1. Los desarrolladores presentan deficiencias en la descripción de los cambios realizados en el versionamiento del código en los sistemas de software.

H2. Los desarrolladores presentan deficiencias en el tipo de cambio ejecutado en los archivos en el versionamiento del código en los sistemas de software.

3. METODOLOGÍA

Apoyando la opinión de Rozo-Rodríguez, Casanovas y Pollo-Cattaneo (2020), es seguro que la importancia en la ingeniería de software promueve la evaluación de las soluciones informáticas en base al nivel de calidad que éstas contienen.

En la revisión del estudio se utilizó un tipo de investigación cuantitativo y cualitativo, con análisis descriptivo de la data; y transversal por ser realizado en un solo tiempo. El instrumento de medición principal fue un script automatizado desarrollado por auditoría propia para extraer y analizar el historial de commits de cada repositorio. Este enfoque

se justifica porque, como menciona Gracia-Orejuela (2022), el componente humano y el esfuerzo manual introducen una alta propensión a errores, requiriendo mucha experiencia para identificar fallas.

Para la recolección de datos, se seleccionó una muestra intencionada de cuatro repositorios de software interconectados, todos componentes críticos del flujo de negocio "Onboarding Digital" dentro del GitHub Enterprise de una institución financiera. La elección de repositorios de código como fuente de datos es una práctica consolidada, ya que, como afirman Chakraborty y Aithal (2022), un repositorio GIT y el seguimiento de cambios lo definen como el lugar de trabajo para almacenar archivos, siendo la base de la colaboración moderna.

El script validó programáticamente cada mensaje de commit contra una expresión regular diseñada para verificar el cumplimiento del estándar de *Conventional Commits*, identificando así la adherencia del equipo a una práctica de documentación estructurada.

4. RESULTADOS

La **Tabla N°1** resume los hallazgos específicos por repositorio, donde aparece el nombre del repositorio, los commits que cumplen con los criterios del *Conventional Commits*, los que no cumplen y, por último, el porcentaje de cumplimiento en cada repositorio:

Tabla N°1. Commits ejecutados en el último trimestre validados por el script.

Nombre de repositorio	Válidos	Inválidos	Porcentaje de cumplimiento
repositorio 1	2	8	20
repositorio 2	10	52	16.13
repositorio 3	3	8	27.27
repositorio 4	300	911	24.77

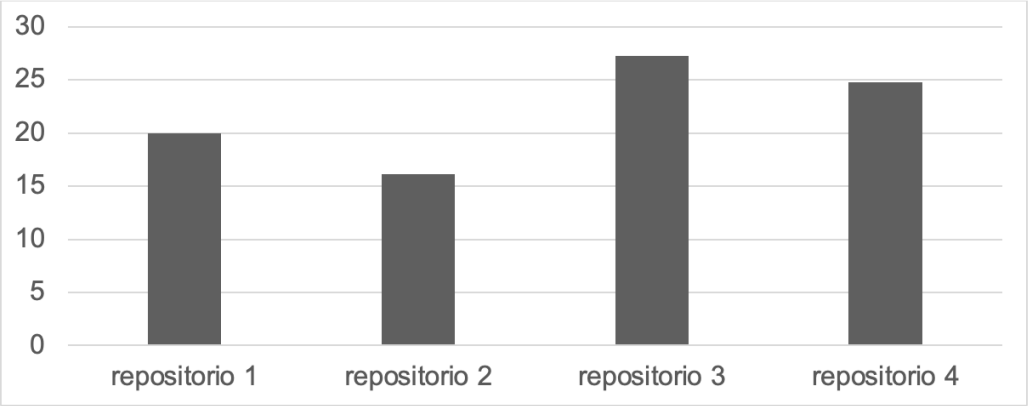
Fuente. Datos tomados de auditoría propia (2025).

Tras ejecutar el script de análisis sobre los cuatro repositorios seleccionados durante el periodo del 1 de mayo de 2025 al 1 de agosto de 2025, se obtuvieron resultados cuantitativos que revelan las prácticas de versionamiento del equipo de desarrollo.

A nivel general, se examinó un total de 1,294 commits. De estos, únicamente 315 cumplieron con el estándar de *Conventional Commits*, mientras que 979 no coincidieron a la especificación. Este análisis arroja un porcentaje de cumplimiento consolidado del 24.34% para el flujo de "Onboarding Digital".

El gráfico de barras de la **Figura N°1**, ilustra de manera comparativa el porcentaje de cumplimiento entre los repositorios analizados, destacando visualmente estas diferencias. Donde las barras muestran el porcentaje de cumplimiento de cada repositorio.

Figura N°1. Gráfica de porcentaje de cumplimiento en los repositorios de Onboarding Digital.



Nota. La figura muestra las cifras de porcentaje de commits en el repositorio de Onboarding Digital en el segundo trimestre del 2025. Fuente: Auditoría Propia (2025).

De estos datos se desprenden varias observaciones clave. En primer lugar, el bajo cumplimiento general del 24.34% evidencia que la mayoría de los commits no siguen la convención establecida.

En segundo lugar, existe una notable variabilidad entre los componentes; el repositorio 3 presenta la mayor adherencia relativa con un 27.27%, mientras que el repositorio 2 muestra la mayor área de oportunidad con solo un 16.13% de cumplimiento, lo que puede indicar diferencias en la cultura o supervisión de los distintos equipos.

5. CONCLUSIONES

La metodología cuantitativa empleada, basada en el análisis automatizado de 1,294 commits, proporcionó una medida objetiva de las prácticas del equipo estudiado. Los resultados fueron contundentes: con una tasa de cumplimiento de apenas el 24.34% respecto al estándar de *Conventional Commits*, se demostró una deficiencia sistemática en la calidad de la documentación de los cambios.

Desde la perspectiva de la gestión de personas, el bajo nivel de cumplimiento (24.34%) evidencia una desconexión entre la capacitación técnica y la cultura organizacional. Este resultado sugiere que los líderes técnicos deben asumir un rol más activo, no solo implementando herramientas, sino fomentando una cultura de disciplina documental. La toma de decisiones gerenciales debe orientarse a establecer políticas claras y métricas de desempeño que incentiven a los desarrolladores a percibir la documentación como una responsabilidad compartida.

Como reflexiona Aponte (2024), la adopción de tecnologías digitales en la industria moderna busca optimizar procesos y reducir la intervención humana en tareas repetitivas; este principio se refleja directamente en los hallazgos de este estudio, donde la automatización del análisis de commits demostró que los equipos aún dependen excesivamente de procesos manuales no estandarizados, lo que subraya la urgencia de integrar herramientas de control de versiones más inteligentes y automatizadas que garanticen la calidad del registro documental.

En la primera hipótesis "Los desarrolladores presentan deficiencias en la descripción de los cambios realizados en el versionamiento del código en los sistemas de software" es aceptada. Esto se evidencia en la baja adherencia al estándar de *Conventional Commits*, lo que genera descripciones ambiguas o incompletas en el historial de versiones.

Como señala Espinoza-Calderón (2025), la falta de un versionamiento adecuado provoca que los miembros trabajen en documentos desactualizados, generando redundancias y perdiendo tiempo valioso. Asimismo, los puentes comunicativos y la transferencia de conocimiento entre equipos técnicos y usuarios finales se facilitan con una documentación clara y efectiva, lo cual no se cumple en este caso.

De igual forma, la segunda hipótesis "Los desarrolladores presentan deficiencias en el tipo de cambio ejecutado en los archivos en el versionamiento del código en los sistemas de software" también es confirmada. El análisis mostró que la mayoría de los commits (979 de 1,294) no especificaban correctamente la naturaleza de la modificación, lo que dificulta el rastreo de errores y la comprensión de la evolución del sistema.

La principal implicación de este estudio es que la eficacia de la documentación no debe ser vista como una formalidad secundaria, sino como un pilar estratégico medible para la salud del proyecto. Para contextualizar la magnitud del hallazgo, los autores Zeng, Zhang, Qiu y Liu (2025) reportan, en el estudio empírico de mayor escala sobre el estándar *Conventional Commits*, que en repositorios que adoptan formalmente el estándar, la tasa de cumplimiento alcanza cerca del 70% de los commits; en contraste, en proyectos que no lo requieren explícitamente, dicha tasa cae a aproximadamente el 10% de forma orgánica.

Frente a estos datos, el 24.34% obtenido en el presente estudio se ubica muy por debajo del umbral esperado para equipos con adopción formal, e incluso apenas supera el nivel de cumplimiento espontáneo, lo que evidencia la ausencia de mecanismos de control documental en el equipo evaluado.

Esta deficiencia puede atribuirse a causas probables como la ausencia de ganchos pre-commit que validen la sintaxis antes de la confirmación, la falta de capacitación formal al incorporar nuevos desarrolladores, y la omisión de la calidad documental como criterio en las revisiones de código.

Entre las causas probables de este bajo cumplimiento se identifican: la ausencia de ganchos pre-commit (pre-commit hooks) que validen automáticamente el formato de los mensajes antes de registrarlos, la falta de capacitación formal en el estándar *Conventional Commits*, y la inexistencia de revisiones de código (code reviews) que incluyan criterios de calidad documental. Del mismo modo, Rozo-Rodríguez, Casanovas, y Pollo-Cattaneo (2020) concluyen que las fases de la ingeniería de software necesitan aumentar su nivel de exigencia ante el crecimiento acelerado de los sistemas, lo que refuerza la urgencia de institucionalizar estas prácticas.

Adicionalmente, los hallazgos reafirman lo planteado por Leones-Zambrano, Macias-Bazurto, Pilla-Zuniga y Fernández-Sánchez (2024) respecto a que un sistema de versionamiento claro es vital para la continuidad del servicio; la alta incidencia de commits mal clasificados (75.65%) detectada en este estudio representa un riesgo directo para la integración de componentes y la compatibilidad de futuras versiones del sistema.

Para revertir estas deficiencias, se proponen tres recomendaciones prácticas específicas. En primer lugar, implementar ganchos pre-commit en los repositorios de la institución que validen automáticamente el formato de cada mensaje antes de permitir su registro, eliminando así la dependencia de la disciplina individual.

En segundo lugar, establecer un programa de capacitación formal en el estándar *Conventional Commits* dirigido a todos los desarrolladores, con métricas de desempeño trimestrales que permitan monitorear la evolución del cumplimiento.

En tercer lugar, incorporar la calidad documental como criterio explícito en las revisiones de código (code reviews), de modo que ningún commit pueda ser integrado a la rama principal sin cumplir el estándar.

Estas medidas, de adoptarse de forma sistemática, permitirían que cualquier miembro del equipo pueda retomar el trabajo de un colega ausente sin demoras, garantizando la continuidad operativa del flujo de Onboarding Digital.

Como trabajo futuro, se proponen tres líneas de investigación con diseño metodológico específico. La primera consiste en replicar este estudio en organizaciones de distintos sectores (salud, retail, gobierno) para validar la generalización de los resultados, utilizando el mismo script de análisis y como métrica principal el porcentaje de cumplimiento del estándar *Conventional Commits*, con el objetivo de establecer benchmarks sectoriales.

La segunda línea propone medir el impacto de la implementación de ganchos pre-commit sobre la tasa de cumplimiento, mediante un diseño cuasi-experimental antes-después en el mismo equipo estudiado, con seguimiento trimestral durante un año.

La tercera línea, de carácter exploratorio, plantea analizar la estructura semántica de los mensajes de commit mediante modelos de lenguaje y grafos de dependencias, con el fin de detectar patrones de error recurrentes y proponer correcciones automatizadas, siguiendo la dirección señalada por Peng, Kalliamvakou, Cihon y Demirer (2023) respecto al potencial de la inteligencia artificial para incrementar la productividad y la calidad en los procesos de desarrollo de software.

Declaración de autoría

Yuliana Sánchez-Acosta: Conceptualización, Análisis formal, Investigación, Metodología, Administración del proyecto, Recursos, Software, Validación, Redacción – borrador original.

María de Jesús Hernández-Garza: Supervisión, Validación, Redacción – revisión y edición.

Referencias Bibliográficas

Acosta-Bravo, E. (2019). Documentación y adopción de buenas prácticas para el desarrollo de software basado en los fundamentos DevOps. Tesis de Grado. Universidad de Antioquia. https://bibliotecadigital.udea.edu.co/entities/publication/d648b494-ca93-4d6d-8ecf-680febd2b19d

Ait, A., Cánovas-Izquierdo, J. y Cabot, J. (2022). *An Empirical Study on the Survival Rate of Github Projects*. 19th International Conference on Mining Software Repositories (MSR '22). Association for Computing Machinery. New York, USA. DOI: https://doi.org/10.1145/3524842.3527941

Álvarez-Fuentes, F. (2025). *Sistema de documentación de desarrollo de software y seguimiento de proyectos*. Proyecto de Graduación. Universidad Tecnológica Centroamericana, Honduras. Centro Universitario Tecnológico CEUTEC. https://repositorio.unitec.edu/items/d3427f49-d5d5-4ff7-b52f-295a1bdfcc53

Aponte, G. M. (2024). Tendencias en la producción científica en el área de tecnologías digitales y economía circular. *Revista Gestión de las Personas y Tecnología, 17*(49). DOI: https://doi.org/10.35588/6436n243

Ávila-Guerrero, F., Bernal-Díaz, I. y Monroy-Gómez, D. (2023). Transformación Digital Empresarial: Revisión de producciones investigativas 2017 – 2021. *Revista Venezolana de Gerencia, 28*(101), 282-296. DOI: https://doi.org/10.52080/rvgluz.28.101.18

Castillo-Rodríguez, J. (2023). Impacto del trabajo remoto en los empleados que viven en el estado de Nuevo León, México. *Revista Gestión de las Personas y Tecnología, 16*(46), 75-93. DOI: https://doi.org/10.35588/5p23qk96

Chakraborty, S. y Aithal, P. (2022). A Practical Approach to GIT Using Bitbucket, GitHub and SourceTree. *International Journal of Applied Engineering and Management Letters (IJAEML), 6*(2). DOI: https://doi.org/10.5281/zenodo.7262771

Conventional Commits. (2023). *Conventional Commits 1.0.0: A specification for adding human and machine-readable meaning to commit messages*. Conventional Commits. https://www.conventionalcommits.org/en/v1.0.0/

Espinoza-Calderón, P. (2025). Competencias desarrolladas en el proceso de optimización de pruebas de software en la empresa Globant. Tesis. Universidad

Continental, Perú. Repositorio Institucional Continental. https://repositorio.continental.edu.pe/handle/20.500.12394/17277

Gracia-Orejuela, K. (2022). *Propuesta de implementación de un proceso de despliegue automatizado, previo pruebas y análisis de código automatizado*. Tesis. Pontificia Universidad Católica del Ecuador, Ecuador. https://repositorio.puce.edu.ec/handle/123456789/38149

Guerrero-Calvache, D. y Hernández, E. (2023). Un estudio exploratorio de las percepciones de productividad en equipos de software ágil. *TecnoLógicas*, *26*(56). 1-19. https://doi.org/10.22430/22565337.2625

Leones-Zambrano, W. Macias-Bazurto, L. Pilla-Zuniga, W. y Fernández-Sánchez, E. (2024). Diseño Estratégico de APIs Escalables y Seguras para la Integración de Sistemas y Aplicaciones. *Ciencia Latina Revista Científica Multidisciplinar, 8*(5), 13395-13408. DOI: https://doi.org/10.37811/cl_rcm.v8i5.14794

Li, J. y Ahmed, I. (2023). *Commit Message Matters: Investigating Impact and Evolution of Commit Message Quality*. IEEE/ACM 45th International Conference on Software Engineering (ICSE). Melbourne, Australia. DOI: https://doi.org/10.1109/ICSE48619.2023.00076

Pastrana-Pardo, M., Ordoñez-Erazo, H. y Cobos-Lozada, C. (2022). *Teaching Guide for Beginnings in DevOps and Continuous Delivery in AWS Focused on the Society 5.0 Skillset*. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, 17*(4). DOI: https://doi.org/10.1109/RITA.2022.3217172

Ma, I. y Lopes, C. (2023). Improving the Quality of Commit Messages in Students' Projects. Preprint. *arXiv preprint arXiv:2304.13887*. DOI: https://doi.org/10.48550/arXiv.2304.13887

Nurlanuly, M., Araz, Z. y Zhuldyz, A. (2025). Analysis the impact of code documentation styles on collaborative software development. *Universum: технические науки*, *5*(134). https://cyberleninka.ru/article/n/analysis-the-impact-of-code-documentation-styles-on-collaborative-software-development

Pachacama-Granda, M., Morales-Cazar, J., Vera-Toscano, E. y Pallo-Morales, J. (2024). *Desarrollo de un prototipo funcional para la gestión del sistema de archivos institucionales EPN utilizando metodologías ágiles*. (Trabajo de Integración Curricular). Escuela Politécnica Nacional, Ecuador. https://bibdigital.epn.edu.ec/handle/15000/25539

Peng, S., Kalliamvakou, E., Cihon, P. y Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. Preprint. *arXiv preprint arXiv:2302.06590.* DOI: https://doi.org/10.48550/arXiv.2302.06590

Hoda, R., Norsaremah, S. y Grundy, J. (2018). The Rise and Evolution of Agile Software Development. *IEEE Software, 35*(5). DOI: https://doi.org/10.1109/MS.2018.290111318

Rozo-Rodríguez, M., Casanovas, I. y Pollo-Cattaneo, M. (2020). Ontología para Transferir Conocimiento en la Etapa de Pruebas de Software. *Revista Gestión de las Personas y Tecnología, 13*(39), 24. DOI: https://doi.org/10.35588/t7krxv27

Slama, F., Ismail, I. y Latrach, L. (2023). Exploring the Integration of Machine Learning Models in Programming Languages on GitHub: Impact on Compatibility to Address Them. Preprint (versión 1). *Research Square*. DOI: https://doi.org/10.21203/rs.3.rs-2591510/v1

Zeng, Q., Zhang, Y., Qiu, Z. y Liu, H. (2025). *A first look at Conventional Commits Classification*. En IEEE/ACM 47th International Conference on Software Engineering (ICSE). Ottawa, Canada. DOI: https://doi.org/10.1109/ICSE55347.2025.00011

Qin, Z., Zhang, H., Qin, X., Xu, K., Dimitrov, K. N. A., Wang, G. y Yu, W. (2018). The Development of Software. En *Fundamentals of Software Culture*. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-13-0701-0_2