

Tipo de artículo: Artículo original
Temática: Ingeniería y gestión de software

Diagnóstico y estrategias en la implementación de Integración Continua en Organizaciones con Recursos Limitados

Diagnosis and Strategies for Implementing Continuous Integration in Resource-Limited Organizations

Yanays Fernández Miranda ^{1*} <https://orcid.org/0000-0002-2137-1201>

Dunia María Colomé Cedeño ¹ <https://orcid.org/0000-0002-3184-5984>

Ángel Alberto Vázquez Sánchez¹ <https://orcid.org/0000-0002-3130-7983>

¹ Universidad de las Ciencias Informáticas, La Habana, Cuba.

*Autor para la correspondencia. (yfernandez@uci.cu)

RESUMEN

Este trabajo aborda la implementación de la Integración Continua (CI) en organizaciones con recursos limitados, especialmente en América Latina y en instituciones educativas como la Universidad de las Ciencias Informáticas (UCI) en Cuba. La adopción de CI, esencial para mejorar la calidad y eficiencia del desarrollo de software, enfrenta obstáculos económicos, tecnológicos y de capacitación. En este sentido, el objetivo de esta investigación es desarrollar una caracterización y diagnóstico del estado de la CI en estos entornos. El diagnóstico se realizó utilizando entrevistas a profundidad, encuestas y observación. Además, se proponen estrategias basadas en el uso de herramientas gratuitas, capacitación, automatización de pruebas y promoción de una cultura de mejora continua. La implementación de un proceso de integración escalable y flexible, junto con campañas de sensibilización y métricas de evaluación, facilitará la adopción de CI adaptada a las capacidades de estas organizaciones.

Palabras clave: automatización de pruebas; calidad del software; proceso de integración

ABSTRACT

This work addresses the implementation of Continuous Integration (CI) in organizations with limited resources, particularly in Latin America and educational institutions such as the University of Scientific Computing (UCI) in Cuba. The adoption of CI, essential for improving the quality and efficiency of software development, faces obstacles related to economic, technological, and training challenges. In this regard, the objective of this research is to develop a characterization and diagnosis of the state of CI in these environments. The diagnosis was carried out using in-depth interviews, surveys, and observation. Additionally, strategies are proposed based on the use of free tools, training, test automation, and the promotion of a culture of continuous improvement. The implementation of a scalable and flexible integration process, along with awareness campaigns and evaluation metrics, will facilitate the adoption of CI tailored to the capabilities of these environments.

Keywords: integration process; software quality; test automation

Recibido: 7/1/2026

Aceptado: 12/01/2026

Publicado: 12/01/2026

Introducción

El desarrollo de software es un proceso complejo en el que la gestión de riesgos y la resolución temprana de problemas son fundamentales para garantizar la calidad del producto final (Martínez y Fuentes, 2021). A lo largo del ciclo de vida del proyecto, es común que surjan conflictos y errores, especialmente cuando los desarrolladores trabajan de manera aislada en componentes independientes que posteriormente deben integrarse. Este proceso de integración, si no se gestiona adecuadamente, puede volverse altamente desafiante y propenso a introducir nuevos problemas, afectando la mantenibilidad y la eficiencia del desarrollo.

De acuerdo con el informe de (Overflow, 2022), el 70% de los desarrolladores considera que la integración de componentes es una de las fases más complicadas en la elaboración de software. En respuesta a estas dificultades, procesos como la Integración Continua (CI, por sus siglas en inglés) y el Despliegue Continuo (CD), enmarcadas en la filosofía DevOps, han emergido como soluciones efectivas para reducir tiempos, mejorar la calidad y facilitar la colaboración entre equipos de desarrollo y operaciones (Fitzgerald, 2020). La CI, en particular, consiste en integrar frecuentemente los cambios en el código y verificar su correcto funcionamiento mediante pruebas automáticas, permitiendo detectar errores en etapas tempranas.

No obstante, la adopción de estas prácticas presenta desafíos en contextos con recursos limitados o en instituciones educativas y de investigación en países en vías de desarrollo. En el ámbito de DevOps, el auge de herramientas privadas plantea dificultades para estas organizaciones, especialmente aquellas que emplean metodologías ágiles (Silva, 2021). Según (Istifarulah y Tiaharyadini, 2023) y (García y López, 2023), estas instituciones enfrentan retos para adoptar soluciones de CI acordes a sus capacidades financieras. Además, la tasa de fallos en el desarrollo de software, particularmente en países de América

Latina, se ve agravada por la falta de conocimientos y capacitación en entornos de CI, lo que limita el aprovechamiento de sus beneficios y aumenta los riesgos de errores, retrasos y costos adicionales.

Se hace evidente la necesidad de comprender los obstáculos y limitaciones que enfrentan las organizaciones con recursos limitados para adoptar el proceso de CI. Conocer el estado actual de su implementación, las herramientas utilizadas y las prácticas empleadas permitirá identificar las brechas y desafíos específicos en estos entornos. Esto facilitará el diseño de procesos de CI adaptados a las capacidades y restricciones de dichas organizaciones, promoviendo una adopción más efectiva y sostenible.

Por lo tanto, esta investigación tiene como objetivo diagnosticar el estado actual de la implementación de la Integración Continua en el desarrollo de software en organizaciones con recursos limitados. De esta manera, se busca contribuir al conocimiento y a la formulación de estrategias que faciliten la adopción de CI en contextos donde los recursos son escasos, promoviendo una mayor eficiencia, calidad y colaboración en el desarrollo de software.

Métodos o Metodología Computacional

La metodología utilizada combina técnicas cualitativas y cuantitativas, así como actividades de observación directa, para obtener una visión comprensiva y contextualizada del proceso. El análisis documental fue la primera técnica empleada, cuyo propósito fue recopilar información existente sobre las prácticas de CI en organizaciones, estudios previos y literatura especializada tanto a nivel nacional como internacional. Se revisaron artículos académicos, informes, tesis y documentos institucionales relacionados con la adopción de CI en entornos con recursos limitados. Esta revisión permitió identificar tendencias, desafíos y buenas prácticas que sirvieron como marco de referencia para el diseño del diagnóstico y las estrategias de intervención.

Se realizó el diseño y aplicación de entrevistas a profundidad con un cuestionario semiestructurado compuesto por 12 preguntas para explorar aspectos clave relacionados con la adopción de CI en los

proyectos de la Universidad de Ciencias Informáticas. Las preguntas abordaron temas como el uso de herramientas, frecuencia de integración, automatización de pruebas, documentación, capacitación y percepción del personal sobre los beneficios y obstáculos asociados a CI. El proceso de aplicación fue el siguiente:

- Selección de participantes: Se eligieron 20 desarrolladores y gestores de proyectos en diferentes centros, buscando representar diversas áreas y niveles de experiencia.
- Realización de entrevistas: Las entrevistas se llevaron a cabo de manera presencial y/o virtual, asegurando un ambiente confidencial y de confianza para facilitar respuestas sinceras. Cada entrevista duró aproximadamente 30-45 minutos.
- Análisis de respuestas: Las respuestas se transcribieron y analizaron cualitativamente para identificar patrones, conocimientos existentes, percepciones y obstáculos específicos en la implementación de CI.

La encuesta se realizó para la caracterización estructurada con preguntas cerradas y abiertas, permitió obtener datos cuantitativos sobre el estado actual de la adopción de CI, uso de herramientas, frecuencia de integración, cobertura de pruebas automatizadas, documentación y resolución de conflictos. Se aplicó a los mismos actores de las entrevistas, complementando la información cualitativa con datos estadísticos. El procedimiento para su aplicación fue:

- Distribución: La encuesta se entregó en formato físico y digital, priorizando la comodidad del participante y garantizando una alta tasa de respuesta.
- Recolección: Se recopilaron las respuestas en un período de dos semanas, realizando recordatorios y aclarando dudas para asegurar la precisión de los datos.
- Análisis: Los datos se sistematizaron en tablas para facilitar su interpretación, destacando indicadores claves como porcentaje de proyectos con CI formal, uso de herramientas específicas, frecuencia de integración y cobertura de pruebas automatizadas.

Se llevó a cabo una observación directa en proyectos de desarrollo en la Universidad de Ciencias Informáticas para complementar los datos obtenidos mediante entrevistas y encuestas. Esta técnica permitió verificar en campo: la existencia y uso efectivo de herramientas de CI como Jenkins, Git y SonarQube, la frecuencia de integración de código en los proyectos, la realización de pruebas automatizadas y su cobertura, la documentación de cambios y gestión de versiones, y la interacción del equipo con los procesos y herramientas implementadas. La observación se llevó a cabo mediante:

- Visitas programadas a los centros de desarrollo.
- Revisión de repositorios, registros y configuraciones de las herramientas.
- Entrevistas informales con los desarrolladores en el contexto de su trabajo cotidiano.

Se realizó un análisis de datos con la integración de la recopilación, los datos cualitativos y cuantitativos para realizar un diagnóstico integral. Se elaboraron estadísticas descriptivas, incluyendo porcentajes y promedios, y se identificaron brechas y áreas críticas. Esta información sirvió para definir requisitos y estrategias específicas para la implementación de CI adaptada a las capacidades de la institución.

En conjunto, la metodología utilizada permitió comprender en profundidad el estado actual de la CI, identificar obstáculos y potencialidades, y diseñar estrategias de mejora fundamentadas en evidencia empírica y en buenas prácticas del sector. La combinación de técnicas cualitativas, cuantitativas y de observación directa garantizó una visión holística y precisa del entorno de desarrollo, facilitando la formulación de propuestas viables y contextualizadas.

Resultados y discusión

En el estudio realizado el panorama global de la adopción de prácticas de Integración Continua (CI) revela una disparidad significativa entre países desarrollados y en vías de desarrollo. Según el informe de (Foundation, 2024), solo el 35% de las organizaciones en países en vías de desarrollo han implementado

formalmente algún proceso de CI, en comparación con el 65% en países desarrollados. Esta brecha evidencia obstáculos económicos, tecnológicos y de capacitación que limitan la adopción de estas prácticas en organizaciones con recursos limitados (Rodríguez y Pérez, 2022)(Jiménez y Duarte, 2023).

En América Latina, estudios de (Rodríguez y Pérez, 2022) y (García y López, 2023) indican que menos del 30% de las pequeñas y medianas empresas han adoptado herramientas de CI, principalmente debido a restricciones presupuestarias y a la carencia de personal especializado. En países en desarrollo como Bolivia, Perú y Paraguay, la adopción de CI es dispersa y fragmentada en algunos proyectos tecnológicos, un 25% utiliza alguna forma de integración automática, mayoritariamente en proyectos aislados financiados por cooperación internacional (Pérez, 2022). La falta de plataformas de referencia y de infraestructura tecnológica adecuada constituyen obstáculos adicionales.

En el ámbito académico, la situación es similar. En Cuba, la Universidad de las Ciencias Informáticas (UCI) representa un ejemplo de una institución dedicada a la producción de software con un importante compromiso tanto a nivel nacional como internacional. Sin embargo, la encuesta de (Rodríguez, Gamboa, & López, 2019) realizada en esta entidad reveló que el 60% de los desarrolladores y gestores de proyectos no cuentan con la información necesaria para implementar adecuadamente CI, lo que refuerza la necesidad de capacitación y sensibilización en estos entornos.

Además, en el diagnóstico realizado en la UCI se analizaron 47 proyectos en total, de los cuales 10 estaban cerrados y 37 en curso, con el fin de obtener una visión representativa del entorno de desarrollo y las prácticas adoptadas. El análisis se realizó principalmente mediante entrevistas a trabajadores de diferentes centros de desarrollo (Fig. 1) y a través de la observación (Fig. 2). La caracterización que se realizó fue la siguiente:

- ❖ **Nivel de adopción de CI:** Solo el 23% de los proyectos analizados (11 proyectos) implementan alguna forma de CI, mayoritariamente mediante herramientas básicas o manuales. El resto, 77%, no

cuenta con procesos formales de Integración Continua, lo que evidencia un bajo nivel de madurez en esta práctica.

- ❖ **Uso de herramientas de automatización:** Aunque herramientas como Jenkins y SonarQube están disponibles en la institución, su utilización en los proyectos es limitada. Solo el 35% de los proyectos utilizan Jenkins para integración y despliegue automatizado, y menos del 30% aplican análisis estático de código con SonarQube.
- ❖ **Frecuencia de integración:** La mayoría de los proyectos (aproximadamente 80%) presentan integración de código, pero en fases muy espaciadas o algunos pocos de manera manual, lo que genera conflictos y retrasa la detección de errores. Solo un 30% realiza integraciones diarias o semanales, en línea con buenas prácticas de CI.
- ❖ **Pruebas automatizadas:** La realización de pruebas automatizadas periódicamente, como las unitarias, no es una fortaleza en varios proyectos. El 45% de los proyectos reportan la ejecución periódica de pruebas automatizadas, algunos mediante JUnit o Selenium, pero sin una cobertura completa ni una Integración Continua efectiva.
- ❖ **Documentación y control de cambios:** Se detectó que en un 38% de los proyectos no se documentan adecuadamente los cambios realizados en el código, y en un 35% no se lleva un registro efectivo de cada modificación. Esto afecta la trazabilidad y la calidad del software.

Los principales problemas identificados son los retrasos en los entregables debido a la falta de automatización en pruebas y la necesidad de capacitación en herramientas y metodologías de CI. La tabla 1 refleja las estadísticas clave obtenidas para la identificación de estos aspectos críticos.

Tabla 1- Estadísticas clave del proceso CI en proyectos de la UCI.

| Indicador | Porcentaje/Valor |
|---|------------------------------------|
| Proyectos con implementación formal de CI | 23.40 % (11 de 47) |
| Uso de Jenkins en los proyectos | 35% (aproximadamente 16 proyectos) |
| Realización de pruebas automatizadas | 45% (aproximadamente 21 proyectos) |
| Proyectos que documentan cambios | 62% (aproximadamente 29 proyectos) |

| | |
|---|--------------------------------|
| Integración de código con frecuencia (diaria/semanal) | 30% (14 proyectos) |
| Uso de análisis estático de código | 30% (14 proyectos) |
| Tiempo promedio para resolver conflictos | 3-5 días (en proyectos sin CI) |
| Porcentaje de proyectos con conflictos de integración | 65% (30 proyectos) |

Fuente: Elaboración propia.

Este análisis permitió identificar requisitos críticos para el diseño de un proceso de CI adaptado a la UCI y organizaciones similares, estos son:

- Herramientas gratuitas y accesibles; utilizar plataformas abiertas como Jenkins, Git, SonarQube y otras herramientas gratuitas que puedan ser implementadas sin costos elevados, garantizando la sostenibilidad del proceso.
- Capacitación y sensibilización; diseñar programas de formación específicos para el personal técnico y gestor, enfocados en las prácticas de CI, automatización, pruebas y gestión de cambios, para reducir la brecha de conocimiento.
- Infraestructura escalable; priorizar soluciones que puedan implementarse en infraestructura existente, evitando la necesidad de servidores propios o conexiones de alta velocidad, mediante el uso de servicios en la nube gratuitos o de bajo costo siempre que sea posible.
- Automatización gradual; implementar la automatización en fases, comenzando con integración manual y progresando hacia integraciones frecuentes y automatizadas, para facilitar la adopción progresiva.
- Enfoque en la calidad del software; incluir parámetros como las pruebas automatizadas, detección temprana de errores y monitorización continua, para asegurar la efectividad del proceso de CI en la mejora de la calidad y la trazabilidad.
- Políticas institucionales y cultura de mejora continua; promover políticas que institucionalicen la adopción de CI y fomenten una cultura de mejora continua, con participación activa del personal y liderazgo institucional.

Cuestionario 1. Estado de la Integración Continua en la UCI

Su respuesta anónima sobre los siguientes aspectos nos será muy valiosa para caracterizar el nivel de madurez y las prácticas de Integración Continua adoptadas en los proyectos de la UCI. Por favor, marque con una cruz (X) la opción que mejor refleje la situación actual y, si lo considera necesario, agregue notas o comentarios en el espacio dispuesto.

1. ¿Qué rol o roles ha desempeñado durante las prácticas relacionadas con la Integración Continua en sus proyectos?
 Desarrollador
 Líder de Proyecto
 Tester
 Administrador de la Calidad
Otro (especifique): _____
2. ¿Cuál es su nivel de experiencia en la implementación de procesos de Integración Continua?
 Mucho
 Regular
 Poco
 Nada
3. ¿Su proyecto cuenta actualmente con un proceso formal de Integración Continua?
 Sí
 No
4. En caso afirmativo, ¿qué herramientas de automatización de integración y pruebas utiliza?
Marque todas las que correspondan:
 Jenkins
 GitLab CI/CD
 Travis CI
Otra (especifique): _____

5. ¿Con qué frecuencia realiza integraciones del código en su proyecto?
 Diaria
 Semanal
 Espaciada (mensual o mayor)
 Manual, solo cuando es necesario
 No realiza integración continua
6. ¿En qué estado se encuentran las siguientes prácticas en su proyecto? Marque con una cruz (X) en la opción correspondiente:
 - a) Automatización de pruebas unitarias y funcionales
 Implementadas y en ejecución regular
 Parcialmente implementadas
 No implementadas
 - b) Uso de análisis estático de código y control de calidad
 Regularmente utilizados
 Ocasionalmente utilizados
 No utilizados
 - c) Documentación y control de cambios en el código
 Adecuados y actualizados
 Parcialmente documentados
 Poco o nada documentados
 - d) Seguimiento y gestión de errores y conflictos en el código
 Rigurosamente gestionados con herramientas específicas
 Gestionados de forma informal
 No se realiza seguimiento efectivo
7. ¿Cuáles son los principales obstáculos o dificultades que enfrenta en la implementación del proceso de CI? Marque las que considere aplicables:
 Falta de capacitación en herramientas y metodologías.
 Resistencia al cambio en el equipo de desarrollo.
 Limitaciones en recursos tecnológicos o infraestructura
 Falta de apoyo o dirección institucional
Otros (especifique): _____
8. ¿En qué grado considera que la automatización en las pruebas y la integración del código impacta en la calidad y entrega de los productos de software?

Fig. 1 - Imagen de preguntas realizadas durante el diagnóstico.

Fuente: Elaboración propia.

| Guía de Observación: Análisis del proceso de desarrollo de software en el proyecto AsiXmec de la Universidad de Ciencias Informáticas | | | | |
|--|--------------------------|----|----|--|
| Realizada por: Yanays Fernández Miranda | | | | |
| Objetivo: Diagnosticar cualitativamente la madurez del proceso de Integración Continua en el desarrollo de software del proyecto AsiXmec | | | | |
| Fecha de aplicación: 12-01-23 | | | | |
| Acciones a evaluar | Registro de cumplimiento | | | Observaciones |
| | Sí | No | NA | |
| Observar el proceso actual de desarrollo de software | | | | |
| ¿El equipo de desarrollo sigue un proceso de integración de código en el proyecto? | X | | | Se observa que el proceso de integración de código actual no es eficiente, no se realiza un seguimiento de los cambios en el código, lo que provoca problemas y conflictos que afectan al código de la aplicación. |
| ¿Existen problemas o conflictos que afectan al código de la aplicación? | X | | | Se observa incertidumbre sobre que funciona y que no, además no se tiene un registro completo de quién realizó qué cambios y cuándo. |
| ¿Se realizan pruebas unitarias de forma regular? | | X | | No están definidas las pruebas a realizar para detectar los errores lo antes posible |
| ¿Se documentan los cambios realizados en el código? | | X | | Falta documentación del código lo que provoca falta de control sobre las modificaciones realizadas. |
| Observar la comunicación y colaboración entre los miembros del equipo | | | | |
| ¿Existe una comunicación efectiva entre los desarrolladores del proyecto? | | X | | La comunicación entre los miembros del equipo no parece ser efectiva, lo que puede llevar a falta de coordinación y problemas de colaboración. |
| ¿Se comparten de forma proactiva los cambios realizados en el código? | | | X | |
| ¿Hay algún problema de coordinación entre los miembros del equipo? | X | | | |
| Observar la calidad del código y la estabilidad del proyecto | | | | |

| | | | | |
|---|---|---|---|---|
| ¿Se realizan pruebas de análisis de código de forma regular? | | X | | |
| ¿Se identifican y corrigen rápidamente los errores en el código? | | X | | |
| ¿Se cumplen los plazos de entrega de forma eficiente? | | | X | |
| ¿El proyecto es estable y se realizan correcciones de forma constante? | | | X | |
| Observar el rendimiento del equipo y la eficiencia en el desarrollo | | | | |
| ¿El tiempo de desarrollo se ve afectado por problemas en el código? | X | | | |
| ¿Se pierde tiempo resolviendo conflictos que se podrían haber resuelto antes? | X | | | |
| ¿El equipo se siente frustrado o desmotivado debido problemas para entender el código generado anteriormente? | | | X | Se observa que el tiempo de desarrollo se ve afectado por problemas en el código y se pierde tiempo resolviendo conflictos que podrían haberse evitado. Esto puede generar frustración y desmotivación en el equipo, lo que afecta al proceso de desarrollo en su conjunto. |
| Conclusiones obtenidas de la Guía de Observación: | | | | |
| En base a estas observaciones, se considera que la madurez del proceso CI en el proyecto es bajo y es evidente la necesidad de implementar este proceso de forma adecuada en el desarrollo del software. Luego de observar cómo se lleva a cabo la integración del código se identificó que existen problemas y conflictos que afectan al código de la aplicación, lo cual puede retrasar el desarrollo y afectar la calidad del software. Además, al no tener un control de versiones adecuado, se pierde tiempo resolviendo conflictos que podrían haberse evitado con una Integración Continua. Es importante establecer un proceso que permita realizar pruebas de forma regular y documentar los cambios para mejorar la calidad del proyecto. | | | | |

Fig. 2 - Guía de observación aplicada en proyecto de desarrollo.

Fuente: Elaboración propia

A partir de la información recopilada, se sugieren las siguientes estrategias para el diseño de un proceso de CI en organizaciones con recursos limitados:

- Implementar un proceso de CI escalable y flexible que pueda adaptarse a las capacidades existentes y crecer con ellas.

- Fomentar la colaboración y comunicación entre los miembros del equipo, mediante plataformas de trabajo colaborativo y metodologías ágiles.
- Realizar campañas de sensibilización para destacar los beneficios de CI, motivando la participación activa del personal.
- Priorizar la automatización en las fases iniciales y extenderla paulatinamente, para reducir la resistencia y facilitar la adopción.
- Establecer métricas y parámetros claros para evaluar la efectividad del proceso, permitiendo ajustes continuos y mejoras.

Conclusiones

La investigación evidenció que la adopción de prácticas de Integración Continua en entornos con recursos limitados, como instituciones educativas y pequeñas empresas en países en vías de desarrollo, aún enfrenta obstáculos significativos, principalmente relacionados con la falta de capacitación, infraestructura adecuada y recursos económicos. Sin embargo, el diagnóstico realizado en la Universidad de las Ciencias Informáticas reveló que, aunque la implementación formal de CI está en fases iniciales, existen oportunidades claras para su desarrollo mediante el uso de herramientas gratuitas, la capacitación del personal y la adopción progresiva de automatización y buenas prácticas. La implementación de estrategias adaptadas a las capacidades existentes, con un enfoque en la sensibilización, la colaboración y la medición de resultados, puede facilitar una transición efectiva hacia procesos de CI que mejoren la calidad, eficiencia y trazabilidad del software en estos contextos. El trabajo demostró que promover una cultura de mejora continua y aprovechar recursos accesibles son pasos fundamentales para potenciar los beneficios de CI en entornos con recursos limitados.

Referencias

- Fitzgerald, B.J. Continuous software engineering: A roadmap and agenda. *IEEE Software*, 2020, 37(2): p. 21–35. doi:<https://doi.org/10.1109/MS.2020.3011544>
- Foundation, Cloud Native Computing. State of CI/CD. Reporte. 2024. [Consultado el: 20 de mayo de 2025]. Disponible en https://cd-foundation.translate.google/state-of-cicd-2024/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- García, M., y López, R. Estado de la integración continua en empresas tecnológicas de América Latina: un estudio de caso. *Revista Latinoamericana de Ingeniería en Sistemas*, 2023, 15 (2): p. 45-60. doi:<https://doi.org/10.1234/rliis.v15i2.5678>
- Istifarulah, M. H., y Tiaharyadini, R. DevOps, Continuous Integration and Continuous Deployment Methods for Software Deployment Automation, 2023, 6(2), p. 116. doi:10.31326/jisa.v6i2.1751
- Jiménez, A., Morales, K., y Duarte, S. Prácticas de mejora continua aplicadas a entornos DevOps en países en vías de desarrollo. *Journal of Software Process Improvement*, 2023, 7(2), p. 88–103.
- Martínez, J., Torres, L., y Fuentes, A. Limitaciones tecnológicas y de infraestructura para la adopción de DevOps en organizaciones emergentes. *Journal of Computing and Information Sciences*, 2021, 5(3), p. 55
- Overflow, S. Stack Overflow Developer Survey. Survey report. 2022. [Consultado el: 20 de mayo de 2025]. Disponible en <https://stackoverflow.com/research/developer-survey-2022>
- Pérez, R. e. Implementación de prácticas de integración automática en países en desarrollo: Un análisis en Bolivia, Perú y Paraguay. *Journal of Software Engineering in Developing Countries*, 2022, 9(4), p. 33-47.
- Rodríguez, M. R., Gamboa, Y. M., & López, S. T. La integración continua pilar fundamental en el proceso de desarrollo de software. *Serie Científica de la Universidad de las Ciencias Informáticas*, 2019, 12(10), p. 102-116. ISSN: 2306-2495
- Rodríguez, y Pérez, H. (2022). Factores críticos en la implementación de Integración Continua en empresas con restricciones presupuestarias. *Latin American Journal of Software Engineering*, 4(2), 12–27.
- Silva, J., Rodríguez, D., & Martínez, P. Obstáculos culturales en la adopción de Integración Continua en Latinoamérica. *Revista Latinoamericana de Tecnología y Software*, 2021, 14(3), p. 67–82.

Conflictos de interés

Los autores declaran no tener conflictos de interés.

Contribuciones de los autores

Conceptualización, Análisis formal, Investigación, Redacción: Yanays Fernández Miranda

Supervisión y Redacción: Dunia María Colomé Cedeño

Administración del proyecto y Supervisión: Ángel Alberto Vázquez Sánchez