

## Calidad 4.0 en Ingeniería de Software: propuesta de un marco de trabajo

Kristian Petkoff Bankoff<sup>[0000-0001-6707-9268]</sup>, Rocío Muñoz<sup>[0000-0001-6521-0568]</sup>, Ariel Pasini<sup>[0000-0002-4752-7112]</sup>, and Patricia Pesado<sup>[0000-0003-0000-3482]</sup>

Instituto de Investigación en Informática LIDI (III-LIDI)  
Facultad de Informática, Universidad Nacional de La Plata, Argentina  
{kpb,rmunoz,apasini,ppesado}@lidi.info.unlp.edu.ar

**Resumen** La Ingeniería de Software es una disciplina en constante evolución que se encuentra presente en todos los aspectos de la vida cotidiana. La calidad del software es un aspecto abordado desde hace tiempo, surgida de los mismos principios de la calidad en la industria manufacturera pero adaptada a las características propias del desarrollo de productos no tangibles. La industria ha evolucionado hacia una revolución digital que implica el uso de tecnologías emergentes como Internet de las cosas (IoT), inteligencia artificial y aprendizaje automático. La Calidad 4.0 adopta los principios de la Industria 4.0 y los aplica a la gestión de la calidad, con un enfoque basado en el análisis de grandes volúmenes de datos, el monitoreo en tiempo real y un mayor enfoque en el cliente mediante el uso de herramientas de comunicación y colaboración. Se propone un marco de trabajo que integra diferentes procesos, herramientas y técnicas de la Ingeniería de Software con el objetivo de habilitar la implementación de principios de la Calidad 4.0 en el desarrollo de software, tomando a DevOps como habilitador al ser un enfoque que promueve la automatización y la integración de herramientas.

**Keywords:** Quality 4.0, Software Engineering, DevOps

### 1. Introducción

El concepto de Calidad 4.0 se refiere a la integración de tecnologías digitales en los sectores industriales y de fabricación para potenciar la toma de decisiones basada en datos, el monitoreo en tiempo real y la integración de tecnologías como Internet de las cosas (IoT), inteligencia artificial y análisis de grandes volúmenes de datos[1].

El desarrollo de software es una parte crítica de la digitalización de la industria que promueve este enfoque [2] ya que es transversal a prácticamente todos los demás principios y características del mismo. Por otra parte, la gestión de la calidad en el proceso de desarrollo de software y en la del propio producto de software es de suma relevancia, lo que se demuestra en la diversidad de estándares disponibles y su aplicación en la industria [3].

En este artículo se propone un marco de trabajo que integra diferentes procesos, herramientas y técnicas de la Ingeniería de Software con el objetivo de habilitar la implementación de principios de la Calidad 4.0 en el desarrollo de software, utilizando como habilitador el concepto de DevOps [4].

El artículo se encuentra organizado de la siguiente manera: en la sección 2 se presentan los conceptos principales de Calidad 4.0, del proceso de Ingeniería de

Software, metodologías ágiles y la cultura DevOps; en la sección 3 se presentan las características que requiere la implementación del framework propuesto; en la sección 4 se describen las herramientas seleccionadas para cubrir los requerimientos propuestos y la forma en que se integraron; a continuación, en la sección 5, se resumen los resultados obtenidos, las conclusiones y los trabajos futuros.

## 2. Calidad 4.0 e Industria 4.0

El concepto de Calidad 4.0 se basa en la cuarta revolución industrial, también conocida como Industria 4.0, que se refiere a la integración de tecnologías digitales en los sectores industriales y de fabricación.

La Industria 4.0 aprovecha la automatización, los dispositivos IoT, la computación en la nube y los sistemas ciberfísicos para crear fábricas y procesos inteligentes y conectados. La Calidad 4.0 adopta los principios de la Industria 4.0 y los aplica a la gestión de la calidad, aprovechando las tecnologías emergentes, el análisis de datos y la transformación digital para impulsar la mejora continua y centrarse en el cliente en varias industrias, adaptable a la de servicios y a la ingeniería de software.

En esencia, la Calidad 4.0 enfatiza la toma de decisiones basada en datos, el monitoreo en tiempo real y la integración de tecnologías como IoT, inteligencia artificial y análisis de big data [5].

Al centrarse en las necesidades del cliente, involucrar a las partes interesadas en el proceso de desarrollo y adaptarse continuamente a los requisitos cambiantes, la Calidad 4.0 tiene como objetivo mejorar la calidad de los productos y servicios, optimizar los procesos e impulsar la innovación en la era digital.

### 2.1 Principios de la Calidad 4.0

Existen diversas propuestas para dividir la Calidad 4.0 en características, principios o ejes. Debido a que el concepto se apoya en la Industria 4.0 y en los procesos de producción de bienes tangibles, se hace a continuación una selección de los principios mejor alineados con la industria del software.

- Toma de decisiones basada en datos: aprovecha el análisis de datos, big data e inteligencia artificial para informar la toma de decisiones. En la industria del software, esto se traduce en el uso de monitoreo en tiempo real, modelos de predicción de errores y datos de comentarios de los clientes para guiar los procesos de desarrollo y optimizar la calidad del software.
- Mejora Continua: fomenta una cultura de mejora continua. Este principio se alinea con la metodología de desarrollo de software ágil, enfatizando el desarrollo iterativo, las retrospectivas y el aprendizaje de los comentarios de los clientes para mejorar los productos de software continuamente.
- Centrado en el cliente: comprender las necesidades y preferencias de los clientes es crucial para los equipos de ingeniería de software. Al involucrar a los clientes en todo el proceso de desarrollo y brindarles experiencias personalizadas, los productos de software pueden satisfacer mejor las expectativas de los usuarios.

- Integración de tecnologías emergentes: para la industria del software, abarcar tecnologías emergentes significa adoptar prácticas DevOps, integración continua e implementación para mejorar la colaboración y agilizar la entrega de software.

### 2.2 Relación entre el software y Calidad 4.0

El software sirve como un habilitador crítico para la implementación de Calidad 4.0 en todas las industrias. Dado que la Calidad 4.0 enfatiza la toma de decisiones basada en datos, el monitoreo en tiempo real y la integración de tecnologías emergentes, el software proporciona la infraestructura necesaria para recopilar, procesar y analizar grandes cantidades de datos. A la vez, permite la conectividad y la interoperabilidad de dispositivos y sistemas en el panorama de IoT, lo que permite el intercambio de datos entre los ámbitos físico y digital. El aprovechamiento de la inteligencia artificial y del aprendizaje automático para identificar patrones, predecir defectos y optimizar procesos en tiempo real aportan mejoras sustanciales a los procesos.

La flexibilidad y adaptabilidad del software son esenciales para implementar prácticas de mejora continua, desarrollo iterativo y enfoques centrados en el cliente. Uno de los principios sobre los que se basa la Calidad 4.0 es el desarrollo de aplicaciones de software ad-hoc para satisfacer las necesidades propias de la organización de manera específica y personalizada. A su vez, los principios y conceptos de la Calidad 4.0 tienen implicaciones significativas para la industria del software. Al aprovechar los datos y el análisis, los equipos de ingeniería de software pueden tomar decisiones informadas, identificar tempranamente defectos y mejorar la calidad general del software.

Por otra parte, la Calidad 4.0 se alinea con las metodologías ágiles, lo que promueve el desarrollo iterativo, la retroalimentación continua del cliente y la colaboración; esto permite que los equipos entreguen productos centrados en el cliente que satisfagan las necesidades cambiantes. Al adoptar enfoques centrados en el cliente e involucrar a las partes interesadas en todo el proceso de desarrollo, la ingeniería de software responde mejor a las necesidades y preferencias de los usuarios, lo que da como resultado mejores productos de software.

La Calidad 4.0 impulsa el desarrollo de software hacia prácticas basadas en datos, fomentando el uso de monitoreo en tiempo real, análisis predictivo y modelos de predicción de errores. Esas prácticas facilitan la supervisión continua de los sistemas de software y fomentan una cultura de mejora y aprendizaje continuos.

Los modelos de desarrollo de software ágiles proponen un proceso liviano que tiene como objetivo reducir los costos, potenciar la comunicación y la coordinación y permitir un enfoque adaptable y flexible ante los cambios en los requerimientos [6]. Estas características están directamente alineadas con principios fundamentales de la Industria 4.0 y sirve de base para la Calidad 4.0. Es por esto que a la hora de proponer un marco de trabajo para implementar el concepto de Calidad 4.0 a la Ingeniería de Software resulta conveniente seleccionar modelos de proceso mejor adaptados desde su concepción. Dos de los modelos ágiles más utilizados son Scrum y XP [7]

La intersección entre la Ingeniería de Software y la Calidad 4.0 puede ser llevada adelante a través de DevOps, que es un conjunto de prácticas, filosofías culturales y herramientas de colaboración que tienen como objetivo mejorar y optimizar el ciclo de

vida del desarrollo de software al fomentar la colaboración entre los equipos de desarrollo de software (Dev) y operaciones de TI (Ops). El objetivo de DevOps es automatizar y optimizar el proceso de creación, prueba, implementación y mantenimiento de aplicaciones de software, lo que da como resultado ciclos de desarrollo más rápidos, calidad mejorada y mayor eficiencia. Para esto define un proceso de integración continua (CI) donde los desarrolladores combinan con frecuencia sus cambios de código en un repositorio compartido. Este enfoque ayuda a identificar los problemas de integración de forma temprana, evitando la acumulación de defectos y reduciendo el riesgo de que los errores lleguen a producción. Al detectar problemas en una etapa temprana, se mejora la calidad general del software. Las pruebas automatizadas, incluidas las pruebas unitarias, las de integración y las de uso de extremo a extremo, se ejecutan automáticamente con cada confirmación de código, asegurando que los cambios no introduzcan regresiones o defectos, llevando así a un software más confiable y de mayor calidad.

El proceso de entrega continua (CD) implica que el software se lance a entornos de producción o ensayo de forma rápida y frecuente. La automatización de los procesos de implementación garantiza la coherencia y reduce las posibilidades de errores humanos, además de posibilitar una mejor experiencia del usuario con respecto a las actualizaciones del software. Por otra parte, el uso de la infraestructura como código (IaC), donde el aprovisionamiento y la configuración de la infraestructura se automatizan y versionan junto con el código fuente de la aplicación, reduce la desviación de la configuración y garantiza que el software se ejecute de manera consistente en diferentes entornos, permitiendo reorientar esfuerzos en otras áreas al mantener entornos de ejecución estables.

La naturaleza de integrar las operaciones de TI con todas las partes interesadas de los procesos de desarrollo requiere de la implementación de prácticas de comunicación mejorada y de responsabilidades compartidas. DevOps fomenta un ciclo de retroalimentación rápido entre desarrolladores, probadores y otras partes interesadas a partir de la automatización de la línea de trabajo y de la disponibilidad de datos. Esta retroalimentación constante permite una rápida identificación y resolución de problemas de calidad, lo que lleva a menores costos por errores y a potenciar la mejora del proceso de desarrollo.

Otro principio clave de DevOps es el monitoreo y análisis en tiempo real, que brinda información sobre el rendimiento del software, los patrones de uso y los posibles problemas, permitiendo a los equipos abordar de manera proactiva las inquietudes relacionadas con la calidad.

Las características descritas anteriormente permiten demostrar que DevOps adopta una cultura de mejora continua, en la que los equipos evalúan y optimizan periódicamente sus procesos. Este enfoque iterativo para el desarrollo y las operaciones contribuye a las mejoras continuas en la calidad del software. La integración continua, la entrega continua, las pruebas automatizadas y la configuración de entornos de ejecución como código permiten controlar y mejorar atributos de la calidad del software además del proceso en sí mismo.

### 3. Propuesta de framework para aplicar Calidad 4.0 en la Ingeniería de Software

La Calidad 4.0 propone un enfoque en el que la utilización de características de la Industria 4.0 se integran en todos los procesos de la organización. Cuando se analiza la aplicación de la Calidad 4.0 en el contexto de procesos de desarrollo de software se debe tener en cuenta que no todos los principios propios de la manufactura de bienes pueden ser trasladados de forma directa a esta disciplina, ya que la Ingeniería de Software no tiene como objetivo producir bienes tangibles, no persigue la producción en serie y tiene un ciclo de vida más allá de la entrega al cliente.

La propuesta de este trabajo se limita a los principios que mejor se adaptan a la industria del software, tomando un subconjunto relevante de los principios aplicables por cuestiones de extensión de la implementación.

- **Trazabilidad:** es una característica común a todas las características y funcionalidades del marco de trabajo que las actividades sean trazables y que el estado de los artefactos sea transparente y visible para todas las partes interesadas. La propuesta involucra el uso de contratos inteligentes para registrar los requerimientos del software y posteriormente toda acción que se provoque sobre los mismos, desde su planificación hasta la entrega, pasando por las diferentes etapas del desarrollo y prueba. El uso de blockchain y contratos inteligentes se propone como habilitador de la Calidad 4.0 por formar parte de los principios de la Industria 4.0.
- **Automatización:** cada una de las herramientas y procedimientos que se seleccionen debe permitir la integración automática por medio de API de manera de posibilitar la ejecución autónoma de los procesos alcanzados por el framework y, en especial, la producción de datos de trazabilidad que sean registrados por medio de contratos inteligentes. Esta característica está alineada con la cultura DevOps que se propone como habilitadora de la Calidad 4.0.
- **Aseguramiento de la calidad:** el abordaje debe realizarse integrando herramientas que permitan la automatización del aseguramiento de la calidad, obteniendo métricas del producto y del proceso por su propia ejecución y no por evaluaciones o inspecciones manuales. Las métricas y los indicadores clave de proceso (KPI) deben poder ser definidas por la organización de manera de que sean específicos para el proyecto y opcionalmente pueden ser adaptadas para satisfacer estándares de calidad como por ejemplo la familia de normas ISO 25000. Es deseable que los parámetros de las herramientas de análisis de calidad del producto sean fácilmente ajustables para permitir su adaptación a los cambios en el contexto.
- **Gestión del ciclo de vida de los requerimientos:** las metodologías ágiles promueven la flexibilidad, adaptabilidad y el uso de procesos más livianos; en este contexto, se debe permitir registrar los requerimientos de una forma ágil pero a la vez coherente y trazable. El ciclo de vida de los requerimientos quedará acotado a la creación, el debate técnico y funcional, la implementación, finalización, prueba y aceptación de cada requerimiento individual.
- **Gestión de la configuración del software:** esta característica consiste en controlar los cambios en el software a lo largo de su desarrollo, basado en el versionado del código, la definición de procedimientos para la integración de los cambios, el

control de entrega del software y el mantenimiento de la consistencia [8].

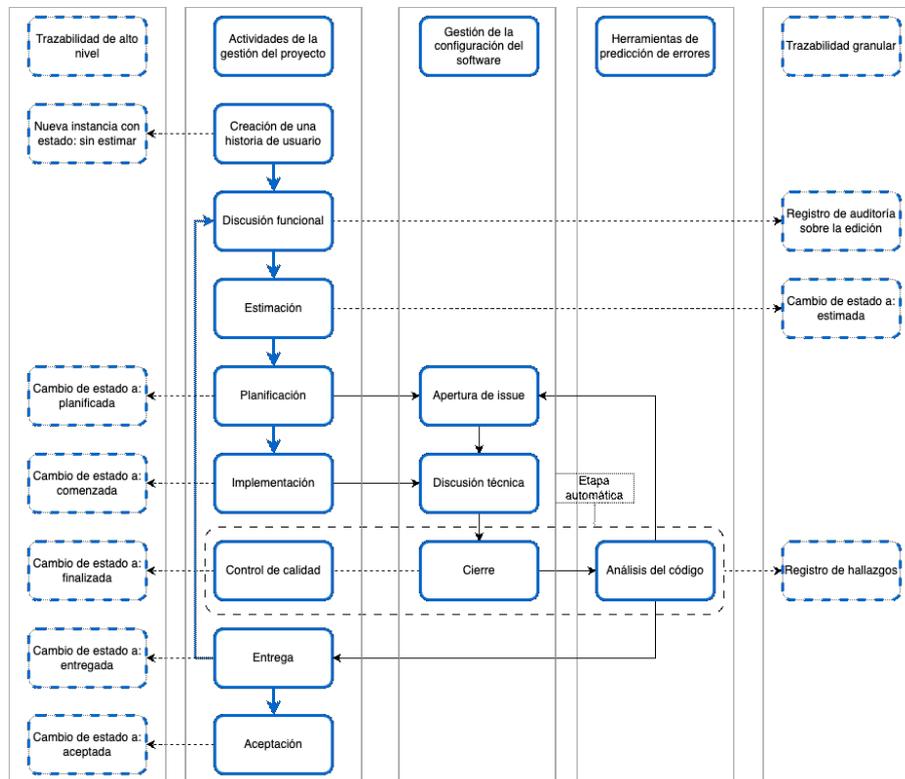
- Integración y adaptación de servicios y visualización de la traza: la integración de diferentes servicios y herramientas para propiciar un registro centralizado y trazable de las actividades del proyecto implican un esfuerzo adicional para sincronizar las diferentes etapas y artefactos y además mantener la coherencia entre los diferentes modelos de datos que utiliza cada aplicación del ecosistema [9]. La información de la traza se debe mostrar de una manera intuitiva al usuario, ya que es primordial involucrar al cliente de extremo a extremo en el proceso. Se definen dos niveles de visualización de trazabilidad: un nivel de planificación, orientado al cliente, y un nivel técnico, con mayor nivel de detalle, de manera que el usuario puede elegir el modo de visualización según sus necesidades.

#### 4. Implementación

El framework propuesto fue evaluado en el contexto de un proyecto llevado adelante por un grupo de alumnos de una asignatura del tercer año de la Facultad de Informática de la Universidad Nacional de La Plata.

El proyecto consistió en el desarrollo de un sistema de gestión de turnos para un centro de vacunación. Los requerimientos del sistema fueron especificados utilizando historias de usuario y la implementación fue llevada adelante en un plazo de dos meses, divididos en tres sprints, basado en el modelo Scrum. La pila de producto y la gestión de los sprints fue realizada con la herramienta Pivotal Tracker. El código fuente del sistema fue alojado en un repositorio de GitLab y la integración continua fue realizada con GitLab CI, mientras que la herramienta de análisis de calidad del código utilizada fue SonarQube.

Para la integración del ecosistema de herramientas se desarrolló un agente ad-hoc que se encarga de la sincronización de los datos entre cada una de estas, ya sea por medio de polling o por medio de webhooks según fuera necesario según las características de cada servicio. La Figura 1 muestra el diagrama de actividades alcanzadas por el framework propuesto.



**Figura1.** Diagrama de actividades alcanzadas por el framework propuesto

Las columnas representan a cada uno de los siguientes componentes:

- **Trazabilidad de alto nivel:** registros de actividad que son de mayor atención para el cliente y para el seguimiento en general del proceso de desarrollo. Se incluyen registros producidos por la aplicación Pivotal Tracker, como la creación de la historia de usuario y los cambios de estado. Esta característica es implementada con contratos inteligentes que registran las transacciones en una red Hyperledger Fabric y una aplicación web de visualización de la traza desarrollada específicamente para este propósito y ajustada al modelo de datos utilizado.
- **Actividades de gestión del proyecto:** corresponde a la creación de la pila de producto mediante la especificación con historias de usuario y discusión a través de comentarios, la planificación de la pila del sprint y el seguimiento del estado de cada requerimiento funcional hasta su entrega en la herramienta Pivotal Tracker, enmarcada en el modelo Scrum.
- **Gestión de la configuración del software:** en esta implementación se integró el versionado de código, el uso de ramas de trabajo y la automatización del proceso de entrega con un seguimiento de las actividades técnicas relacionadas con la implementación desacoplada del seguimiento general del proyecto, permitiendo la traza de las iteraciones dentro del ciclo de implementación de

cada historia de usuario en GitLab manteniendo un estado transparente de cara al cliente.

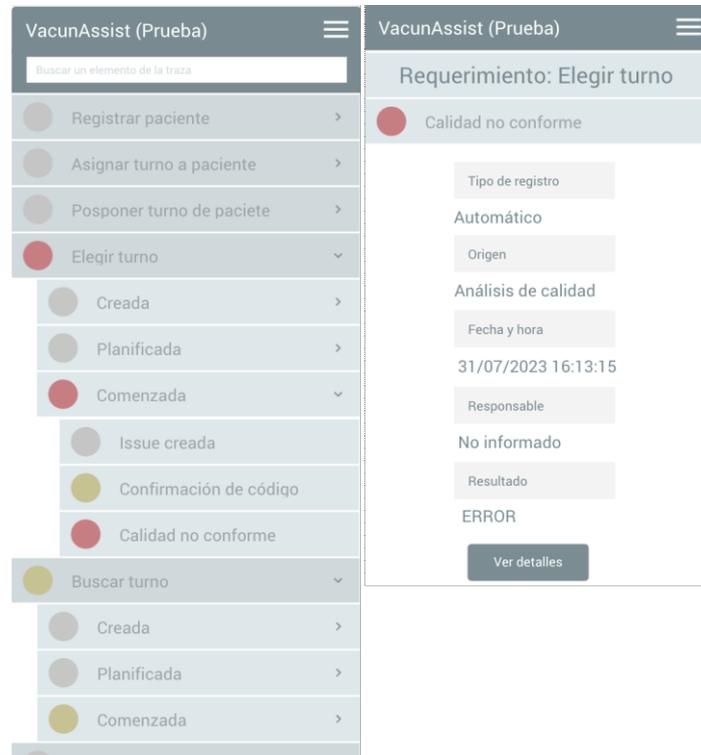
- Herramienta de predicción de errores: la integración de la aplicación SonarQube permitió obtener indicadores de calidad del código a partir del análisis de código ante cada confirmación como parte del proceso de integración continua y de las actividades de la gestión de la configuración del software alcanzadas. Estos indicadores se tuvieron en cuenta para iniciar automáticamente el proceso de entrega de los requerimientos involucrados. En el caso de que los hallazgos no permitan obtener un indicador de calidad aceptable, se genera un registro en la traza y las historias de usuario permanecen en estado de finalización pendiente en la gestión del proyecto. Los parámetros de estos indicadores se pueden configurar para adaptarse a las necesidades del proyecto.
- Trazabilidad granular: elementos de la traza que responden a actividades técnicas con mayor relevancia para el equipo de desarrollo (comentarios en la discusión del requerimiento, estimación de esfuerzo, resultados del aseguramiento de calidad y de los procesos de integración continua y de entrega continua).

La funcionalidad del agente de integración implementado para adaptar los mecanismos de sincronización de las herramientas está representado por las líneas punteadas que abarcan a las actividades de Control de Calidad (Gestión de proyecto), Cierre (Gestión de Configuración del Software) y Análisis de código (Predicción de errores). Este agente es quien se encarga de recibir las actualizaciones de todas las actividades e invocar los contratos inteligentes.

En la Figura 2 (izquierda) se muestra el listado de requerimientos agregados a la traza registrada en la red blockchain implementada con Hyperledger Fabric. Esta aplicación centraliza los datos que se originan y registran en las demás herramientas posibilitando que tanto el cliente como cualquier parte interesada pueda acceder de forma más simple a una mayor cantidad de información, presentada además de manera resumida.

Cada ítem de la traza cuenta con un detalle en el que se puede visualizar información adicional como la fecha y hora del registro, la herramienta que lo originó, el enlace al artefacto (si aplica) y el usuario responsable (en caso de haber sido una acción manual). Un ejemplo de esta visualización se ofrece en la Figura 2 (derecha).

Además de la adaptación de los modelos de datos y de contemplar los mecanismos de sincronización específicos de cada herramienta, se debió tener en cuenta la forma de generar los enlaces a los artefactos, ya que esta información no es proporcionada expresamente por los servicios al emitir un evento. De esta manera, se incrementa el nivel de acoplamiento y dependencia ya que se requieren configuraciones específicas y conocer el esquema de URL y cómo fabricar los enlaces a cada servicio, además de la propia configuración de cada servicio para interactuar con el agente.



**Figura2.** Visualización de los requerimientos en la aplicación de trazabilidad (izq.) y detalles de una transacción (der.)

La situación particular resuelta consistió en configurar Pivotal Tracker para invocar al agente ante actualizaciones en las historias de usuario y, por otra parte, implementar en el agente una consulta activa a SonarQube para obtener datos de los análisis bajo demanda. En el caso de GitLab, la herramienta permite la integración en ambos sentidos mediante webhooks, ejecución de código en el pipeline de integración continua y, también, con acceso por API a los recursos.

Los contratos inteligentes son siempre invocados por eventos externos, por lo que es el agente el que registra la traza. Es importante remarcar que para facilitar esta actividad y garantizar la consistencia de la traza, se optó por evitar la comunicación directa entre los servicios, ya que podría dejar un artefacto en un estado y haber fallado el registro del evento. Se debe velar por la atomicidad entre la traza y la ejecución de la actividad.

## 5. Conclusiones y trabajo futuro

Mediante la implementación del framework de Calidad 4.0 en el contexto de un proceso de Ingeniería de Software, los desarrolladores y el cliente obtuvieron mayor visibilidad de las expectativas de calidad, fomentando la responsabilidad y la

apropiación del producto. El marco proporcionó un mecanismo para realizar un mejor seguimiento del progreso hacia los objetivos y realizar ajustes informados de manera anticipada, con respecto a otros casos testigo. La integración de métricas personalizadas y KPI se adaptó al flujo de trabajo de integración y entrega continua, convirtiéndose en una parte intrínseca de las revisiones de código, las pruebas automatizadas y otros controles de calidad.

Por otra parte, la transparencia en todo el proceso y la disponibilidad de información de traza de manera simple, centralizada e inmediata permitió potenciar la integración del cliente en todo el proceso de desarrollo, incluyéndolo en más etapas y con una frecuencia de comunicación mayor.

La relación entre el software y la Calidad 4.0 es simbiótica y cíclica. El software actúa como un habilitador de la Calidad 4.0 al proporcionar la infraestructura y las herramientas para implementar la toma de decisiones basada en datos y el monitoreo en tiempo real. Al mismo tiempo, muchas herramientas y técnicas de la industria del software se alinean con el concepto de Calidad 4.0, brindando una visión integrada que impulsa mejores prácticas en los procesos de desarrollo, tales como trazabilidad de extremo a extremo de los requerimientos, el enfoque centrado en el cliente, la mejora continua de los procesos, la detección y el tratamiento temprano de errores, etc. Este ciclo refuerza la adopción de los principios de la Calidad 4.0, lo que da como resultado un ciclo de retroalimentación positiva que beneficia tanto a la industria del software como a la implementación de la Calidad 4.0.

A futuro se propone incluir otros principios, como la Inteligencia Artificial, utilizando herramientas como la asistencia para escritura de código o la predicción de errores; también se propone incluir pruebas de integración y de ejecución de extremo a extremo en la implementación propuesta.

### Referencias

1. A review of quality 4.0: definitions, features, technologies, applications, and challenges. *Total Quality Management & Business Excellence* (2021). doi: 10.1080/14783363.2021.1944082
2. Saha, P., Talapatra, S., Belal, H., y Jackson, V.: Unleashing the Potential of the TQM and Industry 4.0 to Achieve Sustainability Performance in the Context of a Developing Country. *Global Journal of Flexible Systems Management* 23(4), 495-513 (2022). doi: 10.1007/s40171-022-00316-x
3. Software quality standards. *Vojnotehnički Glasnik* (2016). doi: 10.5937/VOJTEHG65-10668
4. Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L.E., Tiihonen, J., y Mannisto, T.: DevOps adoption benefits and challenges in practice: A case study. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10027 LNCS, 590 – 597 (2016). doi: 10.1007/978-3-319-49094-6\_44
5. Villegas Forero, D., y Sisodia, R.: *Quality 4.0 -How to Handle Quality in the Industry 4.0 Revolution*. Tesis doct., (2020).
6. Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (ijim)* (2020). doi: 10.3991/IJIM.
7. V14I11.13269

11. Software Development Process Models Comparison And Assessment Of Degree Of Agility Based On Agile Practices And Performance Implementation On XP And Scrum. International Journal of Scientific & Technology Research 8(12), 1008-1016 (2019)
12. PROCEEDINGS
13. Demi, S., Sánchez-Gordón, M., y Colomo-Palacios, R.: A Blockchain-Enabled Framework for Requirements Traceability. En: Yilmaz, M., Clarke, P., Messnarz, R., y Reiner, M. (eds.) Systems, Software and Services Process Improvement, págs. 3-13. Springer International Publishing, Cham (2021)