



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO
PROGRAMA ACADÉMICO DE MAESTRÍA EN INGENIERÍA
DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN

Marco de trabajo DevOps en el proceso de desarrollo de software en
una entidad financiera privada, Lima 2023

TESIS PARA OBTENER EL GRADO ACADÉMICO DE:

Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información

AUTOR:

Moreyra Reyna, Luis Carlos (orcid.org/0000-0002-3792-1415)

ASESORES:

Dr. Vargas Huaman, Jhonatan Issac (orcid.org/0000-0002-1433-7494)

Mg. Puente Zamora, Jonathan Alexis (orcid.org/0009-0007-1034-1617)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y Comunicaciones

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

LIMA – PERÚ

2023

Dedicatoria

Quiero dedicar el presente trabajo a mi familia que a través de sus consejos, enseñanzas y tolerancias me han brindado ese respaldo incondicional para culminar esta etapa de mi vida. También agradecer a mis asesores, gracias a su tiempo y apoyo en brindarme los conocimientos necesarios para el desarrollo de mi formación profesional.

Agradecimiento

Agradezco principalmente a Dios quién me guio y brindó fortalezas para seguir adelante. A mi esposa por animarme a superarme constantemente y que sin su apoyo no hubiera sido posible la elaboración de esta tesis y a todas las personas que de una y otra forma me apoyaron en la realización de este trabajo.



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Declaratoria de Autenticidad del Asesor

Yo, VARGAS HUAMAN JHONATAN ISAAC, docente de la ESCUELA DE POSGRADO MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, asesor de Tesis titulada: "Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera privada, Lima 2023", cuyo autor es MOREYRA REYNA LUIS CARLOS, constato que la investigación tiene un índice de similitud de 18.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 02 de Agosto del 2023

Apellidos y Nombres del Asesor:	Firma
VARGAS HUAMAN JHONATAN ISAAC DNI: 70430225 ORCID: 0000-0002-1433-7494	Firmado electrónicamente por: JIVARGASH el 04- 08-2023 19:35:28

Código documento Trilce: TRI - 0636427





UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Declaratoria de Originalidad del Autor

Yo, MOREYRA REYNA LUIS CARLOS estudiante de la ESCUELA DE POSGRADO del programa de MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, declaro bajo juramento que todos los datos e información que acompañan el Trabajo de Investigación titulado: "Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera, Lima 2023", es de mi autoría, por lo tanto, declaro que el Trabajo de Investigación:

1. No ha sido plagiado ni total, ni parcialmente.
2. He mencionado todas las fuentes empleadas, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes.
3. No ha sido publicado, ni presentado anteriormente para la obtención de otro grado académico o título profesional.
4. Los datos presentados en los resultados no han sido falseados, ni duplicados, ni copiados.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de la información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

Nombres y Apellidos	Firma
MOREYRA REYNA LUIS CARLOS DNI: 45760858 ORCID: 0000-0002-3792-1415	Firmado electrónicamente por: LMOREYRAR el 10-08- 2023 09:52:17

Código documento Trilce: INV - 1234230

Índice de contenidos

Carátula	i
Dedicatoria	ii
Agradecimiento	iii
Declaratoria de Autenticidad del Asesor	iv
Declaratoria de Originalidad del Autor	v
Índice de contenidos	vi
Índice de tablas	vii
Índice de gráficos y figuras	viii
Resumen	ix
Abstract	x
I. INTRODUCCIÓN	1
II. MARCO TEÓRICO	5
III. METODOLOGÍA	13
3.1. Tipo y Diseño de Investigación	13
3.1.1 Enfoque de Investigación	13
3.1.2 Diseño de Investigación	13
3.2. Variables y Operacionalización	14
3.3 Población, Muestra y Muestreo, Unidad de Análisis	16
3.4 Técnicas e Instrumentos de Recolección de Datos	17
3.5 Procedimientos	18
3.6 Método de Análisis de Datos	18
3.7 Aspectos Éticos	19
IV. RESULTADOS	20
V. DISCUSIÓN	35
VI. CONCLUSIONES	39
VII. RECOMENDACIONES	40
REFERENCIAS	41
ANEXOS	48

Índice de tablas

Tabla 1 Tabla de unidad de análisis	17
Tabla 2 Análisis descriptivo para frecuencia de implementación de versiones	20
Tabla 3. Análisis descriptivo para densidad de defectos	23
Tabla 4 Análisis descriptivo para productividad del proceso	26
Tabla 5 Prueba de normalidad para el indicador 1	29
Tabla 6 Prueba de normalidad para el indicador 2	30
Tabla 7 Prueba de normalidad para el indicador 2	30
Tabla 8 Resultados obtenidos de las pruebas de normalidad	31
Tabla 9 Resultados de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones	31
Tabla 10 Resumen de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones	32
Tabla 11 Resultados de la prueba de U-Mann Whitney para el indicador densidad de defectos	32
Tabla 12 Resumen de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones	33
Tabla 13 Resultados de la prueba de U-Mann Whitney para el indicador densidad de defectos	33
Tabla 14 Resumen de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones	34

Índice de gráficos y figuras

Figura 1 Estudio pre-experimental pre-post	14
Figura 2 Comparación de las medias para frecuencia de implementación de versiones	21
Figura 3 Gráfico de cajas para frecuencia de implementación de versiones	22
Figura 4 Comparación de las medias para densidad de defectos	24
Figura 5 Gráfico de cajas para densidad de defectos	25
Figura 6 Comparación de las medias para rendimiento del proceso	27
Figura 7 Gráfico de cajas para densidad de defectos	28

Resumen

La investigación tuvo como objetivo Determinar la influencia del marco de trabajo DevOps en el Proceso de Desarrollo de Software en una Entidad Financiera, midiendo los resultados del desarrollo de proyectos de manera tradicional y utilizando el marco DevOps en los indicadores frecuencia de implementación de versiones, densidad de defectos y la productividad del proceso.

La investigación es aplicada con enfoque cuantitativo y diseño preexperimental del tipo pre y post, la muestra lo conformaron 87 requerimientos, divididos en dos muestras independientes, los datos se obtuvieron utilizando fichas de registro. Para validar las hipótesis se determinó la normalidad de datos utilizando la Z de Kolmogórov-Smirnov para cada indicador, con los resultados obtenidos se aplicó el test no paramétrico de U-Mann Whitney.

Los resultados arrojaron que se logró incrementar la frecuencia de implementación de versiones en 55.17%, igualmente se logró reducir la densidad de defectos en 16% y por último se incrementó la productividad del proceso en 33.33%. Concluyendo que la implementación del Marco de Trabajo DevOps influye positivamente en el Proceso de Desarrollo de Software en una entidad financiera.

Palabras clave: DevOps, Marco de trabajo, Desarrollo de Software y Enfoque Ágil.

Abstract

The objective of the research was to determine the influence of the DevOps framework in the Software Development Process in a Financial Institution, measuring the results of project development in a traditional way and using the DevOps framework in the indicators frequency of version implementation, density of defects and the productivity of the process.

The research is applied with a quantitative approach and pre-experimental design of the pre and post type, the sample was made up of 87 requirements, divided into two independent samples, the data was obtained using registration sheets. To validate the hypotheses, the normality of the data was determined using the Kolmogórov-Smirnov Z for each indicator, with the results obtained the non-parametric U-Mann Whitney test was applied.

The results showed that it was possible to increase the frequency of version implementation by 55.17%, it was also possible to reduce the density of defects by 16% and finally the productivity of the process was increased by 33.33%. Concluding that the implementation of the DevOps Framework positively influences the Software Development Process in a financial institution.

Keywords: DevOps, Framework, Software Development and Agile Approach.

I. INTRODUCCIÓN

El proceso de desarrollo de software, en los últimos años, ha experimentado una evolución significativa gracias a la adopción de nuevas tecnologías y metodologías de trabajo. Sin embargo, aún persisten desafíos y problemas que enfrentan tanto los desarrolladores como las organizaciones involucradas en dicho proceso. Uno de los principales desafíos se encuentra en la escasa interacción y cooperación entre los equipos de desarrollo, lo que puede generar problemas de integración y retrasos en el proyecto. A su vez, el aumento en la complejidad en el desarrollo de sistemas y su dependencia de múltiples componentes pueden dificultar la identificación y resolución de errores y problemas en el software.

En la actualidad, Dhaya (2021) nos menciona que todas las industrias, desde las finanzas hasta la fabricación minorista, dependen del software. Debido a la naturaleza intangible y complicada del software, los requisitos cambian rápidamente y los equipos de desarrollo tienen dificultades para generar software que satisfaga adecuadamente las expectativas del cliente y que, al mismo tiempo, proporcione la funcionalidad y calidad de software deseado. De igual manera Pilliang & Munawar (2022) consideran que los proyectos de software suelen ser complejos y son particularmente susceptibles al fracaso. La mayoría de estos proyectos superan el presupuesto y finalizan prematuramente o no cumplen con las expectativas de los usuarios y las funcionalidades comerciales. A su vez Sankhe et al. (2022) argumentan que, en este escenario, las industrias que desarrollan software han estado implementando recientemente metodologías ágiles para gestionar proyectos en lugar de las tradicionales, debido a que generalmente se consideran pesadas, a diferencia de las ágiles, cuyo objetivo es proporcionar enfoques ligeros a los proyectos. Además, las metodologías ágiles se centran en gestionar y acelerar las actividades de desarrollo. Como señalan Hammad & Inayat (2018), el framework Scrum es el más ampliamente empleado entre los marcos de trabajo y metodologías ágiles en la gestión de proyectos de software. Presenta un conjunto de excelentes preceptos orientados a la pronta entrega de valor a los clientes, al tiempo que logra simplificar el proceso laboral, disminuir el tiempo de desarrollo y realzar la transparencia organizacional.

A nivel internacional Gomber et al. (2018) indica que el desarrollo del software en financieras, ha recibido gran atención de académicos y profesionales debido a su potencial para transformar digitalmente las redes de las cadenas de suministro en casi

todos los sectores empresariales. Para Chen et al. (2019) la financiación global de proyectos de software ha aumentado rápidamente de \$ 38,1 mil millones para todo 2017 a \$ 57 mil millones en 2018. Además Zavolokina et al. (2017) nos explican que el software financiero se basa en la tecnología de la información (TI) para ofrecer productos y servicios financieros dentro de la industria bancaria, con herramientas más avanzadas de gestión de riesgos, procesamiento comercial, gestión de efectivo y análisis de datos implementadas por las instituciones financieras. A su vez Papazafeiropoulou & Spanaki (2016) indican por una parte que el sector financiero haya aplicado tecnologías disruptivas (por ejemplo: Blockchain, Data Analytics, etc.) para transformar los modelos comerciales existentes y desarrollar nuevos productos (por ejemplo, pagos sin efectivo, asesores automáticos, etc.) en la industria de los servicios financieros. Finalmente Gozman et al. (2018) mencionan que más importante aún, el software financiero ofrece confianza y transparencia para los sistemas y transacciones en un campo donde estos aspectos son más requeridos.

Luego Donnelly (2018) menciona el caso de Barclays, una empresa de servicios financieros con sede en Londres que se enfrentaba a una creciente presión normativa y de mercado ejercida por los líderes de los servicios digitales modernos. Con el objetivo de mejorar la innovación y la eficiencia, Barclays decidió implementar una plataforma de servicios para sus aplicaciones, lo que le permitió modernizar todo el soporte tecnológico de los diferentes sistemas de información. Este cambio permitió que sus desarrolladores tuvieran capacidades de autoservicio bajo demanda. Se buscaba reducir los tiempos de despliegue de semanas a horas, para que permita al personal de tecnologías de información dedicarse a proyectos más innovadores y valiosos en lugar de realizar tareas rutinarias.

A nivel nacional Suárez Barcia (2021) manifiesta que el número de empresas financieras de tecnología está aumentando constantemente. A finales de 2020, había 148 empresas financieras de tecnología en todo el país con una tasa de crecimiento anual promedio del 21 %. Asimismo, Perú cuenta actualmente con la Asociación Peruana Fintech, cuyos fundamentos encuentran su raíz en la innovación y la mejora en el uso de los productos y servicios financieros a través del desarrollo de software.

Por otro lado Urresti et al. (2019) señalan que la mayoría de compañías de desarrollo de software tienen menos recursos humanos y menos dinero, por lo que la ausencia de estándares o marcos de desarrollo de software conduce a la incapacidad

de cumplir y establecer estimaciones de tiempo y recursos. En el caso de las PYMES, les es difícil relacionar los estándares de desarrollo de software con las necesidades del negocio, y respecto a la implementación final, en su mayoría no es atractiva para los clientes del negocio.

A nivel local, contamos con el caso de una entidad financiera con gran presencia en la banca peruana, que viene brindando sus servicios desde el año 1987. En la actualidad, se trata de una de las entidades financieras más destacadas del país, con un enfoque en ofrecer productos innovadores y un servicio eficiente y conveniente, a una amplia base de clientes que supera los 2 millones de clientes. La entidad ha enfrentado una serie de problemas digitales en los últimos años, incluyendo fraudes en línea y ataques cibernéticos, problemas de navegación en línea y dificultades en la implementación de nuevos productos y servicios digitales. La institución ha trabajado duro para abordar estos problemas y mejorar la experiencia del cliente en línea, pero aún enfrenta desafíos constantes a medida que el entorno financiero continúa evolucionando. La institución utiliza diversas estrategias, entre ellas la aplicación de metodologías o marcos de trabajo que contribuyen a perfeccionar todas las actividades relacionadas en el desarrollo de software.

La problemática expuesta da lugar al problema general que abarca la investigación: ¿Cómo impacta el Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera? Del cual se desprenden los problemas específicos (i) ¿Cómo impacta el Marco de trabajo DevOps en la implementación de software en una entidad financiera?; (ii) ¿Cómo impacta el Marco de trabajo DevOps en la calidad del producto en una entidad financiera?; (iii) ¿Cómo impacta el Marco de trabajo DevOps en la productividad del proceso en una entidad financiera?

En la investigación, el objetivo general es establecer la influencia del marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera. Asimismo, se presentan objetivos específicos consecutivos, a saber: (i) determinar la influencia del Marco de trabajo DevOps en la implementación de software en una entidad financiera; (ii) determinar la influencia del Marco de trabajo DevOps en la calidad del producto en una entidad financiera, y (iii) determinar la influencia del Marco de trabajo DevOps en la productividad del proceso en una entidad financiera.

Mientras que la hipótesis general es el marco de trabajo DevOps influye significativamente en el proceso de desarrollo de software en una entidad financiera. La

primera hipótesis específica es marco de trabajo DevOps influye significativamente en la implementación de software en una entidad financiera. La segunda hipótesis específica es marco de trabajo DevOps influye significativamente en la calidad del producto en una entidad financiera. La tercera hipótesis específica es marco de trabajo DevOps influye significativamente en la productividad del proceso en una entidad financiera.

Según Bernal (2013), argumenta que una investigación adquiere una justificación práctica cuando su ejecución no solo conlleva a la solución de un problema, sino que además propone estrategias cuya implementación sería de gran utilidad para abordarlo de manera efectiva. De lo anterior se tiene como justificación práctica para la presente investigación en que se aplica un marco de trabajo que permita acelerar y mejorar la calidad en los desarrollos ejecutados con el propósito de abordar las problemáticas y requerimientos que surgen en el ámbito de desarrollo de una entidad financiera.

De acuerdo con Bernal (2013) , la justificación metodológica de una investigación se evidencia cuando el proyecto en cuestión plantea la introducción de un nuevo método o estrategia con el propósito de generar conocimiento válido y confiable. Por consiguiente, el trabajo presente se justifica metodológicamente considerando la exploración del impacto derivado en la adopción del marco DevOps sobre el proceso actual de desarrollo de software de una entidad financiera, lo cual podría resultar provechoso para investigaciones venideras.

La justificación económica en la investigación, se deslinda al validar el impacto de aplicar el marco de trabajo DevOps sobre el ciclo de vida del desarrollo de software. De obtener resultados positivos, se estaría mejorando el proceso actual, lo que se convierte en una disminución de los costos en términos de tiempo y recursos, lo que puede brindar un resultado positivo en la rentabilidad de la entidad financiera y, por ende, un impacto económico relevante

II. MARCO TEÓRICO

En el ámbito nacional, en una investigación realizada por Peralta (2019), en el Ministerio de Cultura, se buscó demostrar que la aplicación de DevOps tiene una influencia positiva en la Entrega Continua de software. El objetivo del estudio se centró en los problemas experimentados en los pases de aplicaciones web a producción, y se encontró que el marco DevOps incrementa la frecuencia de despliegue, reduce los cambios con errores y aumenta los cambios sin fallas. El estudio se realizó empleando un diseño experimental de tipología preexperimental, y se analizó el antes y después de las variables expuestas al estímulo. La comprobación de hipótesis se realizó utilizando la prueba paramétrica de la T de Student. Como resultado del estudio, se constató que el implementar DevOps en el proceso de desarrollo de software produjo una notable reducción en la incidencia de cambios fallidos en un 57.42%, aumento de la frecuencia de despliegue en un 49.26% y aumentar los cambios sin errores en un 25.6% en la Entrega Continua.

En su investigación, Casas (2020) examinó a una empresa del sector de seguros, la influencia de DevOps en el proceso de despliegue e integración de código de software. La hipótesis planteada fue que la implementación de DevOps mejoraría la eficacia de este proceso. Se utilizó un diseño preexperimental y se llegó a la conclusión de que la implementación de DevOps permitió aumentar la cantidad de veces que se libera código de 1,63 a 3,25, reducir el tiempo de ciclo (Cycletime) de 18,11 a 11,61 días y aumentar la ratio de despliegues exitosos de 54,58% a 72,92%. Se concluye que la implementación de DevOps contribuyó de manera significativa a potenciar tanto la velocidad como la calidad del proceso de despliegue e integración de software.

Por su parte Muñoz et al. (2021) proponen un modelo que permite integrar DevOps en los procesos para desarrollar software, considerando tanto aspectos técnicos como organizacionales. El modelo fue testeado en una empresa que provee desarrollo ágil de software y los resultados fueron alentadores, lograron aumentar la frecuencia de despliegue de forma semanal a diaria, y con 16 de los 20 despliegues realizados puestos exitosamente en producción. Esto evidencia que la implementación de DevOps puede elevar la productividad de las organizaciones tienen como objetivo el desarrollo de software. Además, se aborda la discusión acerca de cómo la implementación de DevOps puede potenciar tanto la productividad como la competencia que tienen las empresas de software en un entorno empresarial globalizado.

En su investigación, Castro & Espitia (2022) señalan que la eficacia de la ejecución de un proyecto de software depende de alcanzar las metas establecidas con anticipación. Es esencial llevar a cabo un seguimiento del progreso del proyecto para evaluar su estado y determinar los controles necesarios para implementar las acciones correctivas correspondientes. Con el propósito de resolver las dificultades asociadas a la evaluación del progreso en el ámbito de generación de software, se propone en el escrito el empleo de elementos pertenecientes al enfoque de SEMAT (Sistema de Teoría y Método de la Ingeniería de Software) y de los procesos correspondientes al método QFD (Implementación de la Función de Calidad). El informe detalla diversas estrategias para evaluar el progreso de la creación de programas informáticos y exhibe un esquema de ilustración fundamentado en SEMAT, así como un esquema de confirmación basado en la matriz de excelencia de QFD. Este documento representa una contribución significativa al área de ingeniería de software, dado que habilita directrices a modo de evaluación en gestión de proyectos de software.

En su estudio Suescún-Monsalve et al. (2021), se realizó un estudio exhaustivo y una revisión coherente de la literatura centrada en el adoptar prácticas DevOps para soluciones vinculadas a la Industria 4.0. Los investigadores resaltaron que DevOps, como acrónimo, engloba la integración sinérgica de los ámbitos de desarrollo y operaciones en el área de ingeniería de software, facilita una colaboración efectiva y se centra en aspectos como del ciclo de vida de software como desarrollo, integración, entrega, liberación, pruebas y supervisión continua mediante la automatización de tareas. Los investigadores concluyeron que la adopción de DevOps se vuelve fundamental para las organizaciones que desean afrontar los desafíos inherentes a la Industria 4.0, ya que les brinda la capacidad de desarrollo de productos y servicios de forma ágil y eficiente, reduciendo costos y mejorando los niveles de servicio y gestión. La investigación resalta la importancia y pertinencia de implementar DevOps en el ámbito de la generación de soluciones para la Industria 4.0, por lo tanto, resulta imperativo conceptualizar y adoptar un modelo de desarrollo de soluciones con el propósito de abordar los desafíos surgidos en la revolución industrial indicada y prosperar en la era digital.

En los antecedentes internacionales se tiene a Pastrana-Pardo et al. (2022) que, en su artículo sobre el acercamiento de las mejores prácticas para desarrollar software en base a DevOps, plantean como objetivo principal, el proponer un modelo en el que se considere el versionado de código, integración de código y despliegue continuo en el

desarrollo de software. Las prácticas propuestas en el modelo, generan un ambiente de calidad preventiva que implementa controles significativos para el proceso de desarrollo, también se propone el ejercer integración y despliegue continuo (CI/CD) lo que brindará a los equipos de desarrollo, el asegurar despliegues efectivos. Estas prácticas se evaluaron en 3 empresas distintas, recolectando datos de incidencias reportadas semanalmente y número de despliegues exitosos. Como resultado se obtuvo una reducción del 41.5% en promedio en el número de incidencias reportadas y se logró un 100% en la cantidad de despliegues exitosos. Se concluye que se redujo la cantidad de errores que se presentan luego de realizar un despliegue y se mejoró en la efectividad de los pases a producción.

También en la investigación ejecutada por Marijan et al. (2018), nos presentó una propuesta que planteaba un enfoque destinado a reducir los extensos tiempos de ciclo en el ámbito de DevOps mediante la disminución de la duración de los ciclos de prueba en CI. Dicho enfoque se fundamenta en la aplicación de la optimización de pruebas, tomando en cuenta tanto el historial como el riesgo, con el objetivo de calcular un conjunto de pruebas optimizado que sea capaz de reducir el tiempo total de ejecución de las pruebas, sin comprometer de manera significativa la efectividad en la detección de fallos ni la cobertura de riesgos que los conjuntos de pruebas ofrecen. Este enfoque fue sometido a evaluación experimental y los resultados obtenidos revelaron una mejora significativa en comparación con las prácticas habituales y la selección aleatoria de pruebas. En resumen, los resultados obtenidos respaldan la efectividad del enfoque propuesto para reducir los tiempos de ciclo en DevOps mediante la disminución del tiempo de prueba en CI. Se ha demostrado que el enfoque es altamente efectivo en comparación con la práctica común de volver a probar todo el impacto. Los resultados revelaron una reducción significativa en el tiempo de ciclo de CI de hasta un 35%. Además, se observó que el enfoque logró minimizar el impacto en la efectividad de la detección de fallas y la cobertura de riesgos de las suites de pruebas optimizadas, con disminuciones de hasta un 3% en ambos aspectos. Estos hallazgos confirman que la implementación suma valor y que es recomendable este enfoque para mejorar los procesos de desarrollo software a través de DevOps.

A su vez Ali et al. (2020) llevó a cabo un experimento controlado en el cual dos grupos llevaron a cabo un proyecto piloto con el fin de validar su proceso DevOps híbrido propuesto. Mediante dicho proyecto piloto, se evaluó la eficacia y eficiencia del enfoque propuesto. Se pudo observar que el Grupo 1 hizo uso del 30,6% del total de componentes

reutilizables presentes en el repositorio y logró obtener una mejora promedio del 35,2% en términos de rendimiento del proceso. Por otro lado, el Grupo 2 llevó a cabo el proyecto piloto sin considerar los principios de DevOps híbridos (es decir, sin aplicar una reutilización sistemática) y obtuvo una mejora promedio del 10% en términos de rendimiento del proceso. Los resultados obtenidos del experimento controlado indican que el proceso DevOps híbrido propuesto demostró ser efectivo y eficiente en su funcionamiento.

También Chugh et al. (2021) menciona que la integración del intercambio de conocimientos es de gran importancia en mejorar los procesos de software (MPS). A pesar de ello, son muy pocos los estudios empíricos que han evaluado cómo y en qué medida el intercambio de conocimientos afecta los distintos aspectos de la MPS. Apoyándose el enfoque basado en recursos y conocimientos de las organizaciones de desarrollo de software, se han explorado los antecedentes y las consecuencias del intercambio de conocimientos para el éxito de la MPS. Este estudio desarrolla y valida empíricamente un modelo que presenta cómo el intercambio de conocimientos impacta el éxito de la MPS, cómo factores organizacionales (liderazgo y confianza) y un factor individual (motivación) afectan el intercambio de conocimientos y cómo el usar tecnologías de información facilita explícitamente el vínculo entre el intercambio de conocimientos y el éxito de la MPS. La prueba empírica del modelo se realizó usando la técnica estadística de modelado de ecuaciones estructurales mediante el análisis del conjunto de datos de 556 encuestados recopilados de organizaciones de desarrollo de software de la India. Los resultados revelan que el intercambio de conocimientos mejora los procesos de software en términos de efectividad del diseño, menor densidad de fallas y mejora de la competencia en desarrollo de software. Además, los resultados obtenidos, nos demuestran que al usar la tecnología de la información. Esto modera la relación entre el vínculo de conocimientos y el MPS. Se concluye que los procesos que involucran desarrollar software, deben ser colaborativos e implica a miembros del equipo con diversas habilidades, como especialistas técnicos y estrategas, que trabajan en grupo para el diseño, desarrollo e implementación de nuevos sistemas. El éxito se determina por la captura de conocimiento, intercambiar información y la reducir las interrupciones en la comunicación entre los equipos que desarrollan software. Para aprovechar la competencia entre los equipos que desarrollan software y mejorar la comunicación adaptada a las necesidades de las partes interesadas, los miembros del equipo deben ser capacitados para encontrar medios efectivos de intercambio de conocimiento. Sin

embargo, el intercambio de conocimiento presenta desafíos de calidad de información y ventajas competitivas que pueden resultar en una falta de intercambio de conocimiento incompleto, incorrecto y retrasado.

Por otro lado Lee et al. (2018) en su investigación menciona también que la mejora de procesos de software (MPS) es un continuo aprendizaje organizacional con el objetivo de generar una ventaja competitiva en entornos de negocios y software en constante evolución. Sin embargo, no se ha investigado la capacidad para aumentar la adquisición y utilización efectiva de conocimientos externos de MPS. Por lo tanto, este estudio utiliza la teoría de capacidad dinámica para investigar empíricamente la capacidad de una empresa para absorber conocimientos externos y lograr MPS. Luego propusieron un modelo de investigación, para examinar las relaciones entre la capacidad de absorción de potencial (CAP) de una empresa, la capacidad de absorción realizada (CAR), el éxito de MPS y el rendimiento de la empresa. En este sentido, se encuestaron a 108 encuestados en 56 empresas taiwanesas con certificación MPS. Luego, probaron su modelo utilizando una técnica para modelar que permite estimar el efecto y las relaciones entre las variables. Al revisar los resultados que se obtuvieron, estos nos indican que el éxito de MPS está significativamente influenciado por CAP a través de CAR. De igual manera, los resultados sugieren que la capacidad de absorción es fundamental para el éxito de MPS y que este último influye en última instancia en el rendimiento de la empresa.

En cuanto a los fundamentos teóricos de la investigación, Narang & Mittal (2022) sostienen que la ingeniería de desarrollo de software constituye una disciplina que involucra todo lo necesario del ciclo de vida para desarrollar software (SDLC), el cual comprende diversas etapas que van desde la recolección de requisitos y el análisis exhaustivo, el diseñar, codificar, definir las pruebas y ejecutarlas, realizar la implementación y al final realizar el mantenimiento por cierto tiempo.

Teniendo en cuenta a H. Lewis (2021), manifiesta que las metodologías tradicionales para desarrollar software, incluyen específicamente el modelo de cascada, introducido por primera vez en la década de 1970. El modelo en cascada se explica etapa por etapa. A través de estas etapas, incluidos los enfoques iterativos, el autor introdujo el desarrollo de software a gran escala. Luego Patanakul & Rufo-McCarron (2018) nos indican que el enfoque paso a paso de las metodologías tradicionales permite que los proyectos se manejen y desarrollen con mayor éxito. Pero, por otro lado, los métodos de desarrollo

tradicionales son rígidos y tampoco responden a las solicitudes agresivas o que cambian con frecuencia de los clientes.

Por otro lado Ciric et al. (2019) revelan que, en comparación con la metodología tradicional, la metodología de desarrollo ágil proporciona un conjunto de prácticas que permiten una rápida adaptación a las necesidades cambiantes del cliente. En cuanto a las lagunas de investigación, los autores discutieron muy bien las fallas de los modelos tradicionales, pero no propusieron ninguna solución para superar las principales fallas de las metodologías de desarrollo tradicionales.

Aunque Hobbs & Petit (2017) expresan que se crearon métodos ágiles para abordar cualquier falla o limitación de las metodologías de desarrollo tradicionales, también Dingsøyr et al. (2018) señalan que se ha confirmado que son efectivos para proyectos distribuidos a gran escala. Los enfoques ágiles son impulsores de un desarrollo enorme y extremadamente grande. De acuerdo con Agrawal et al. (2016), los enfoques ágiles fomentan una mayor comunicación, integración continua y entrega rápida de productos utilizando un enfoque incremental e iterativo, pero también tienen varias desventajas, como falta de planificación, falta de documentación, falta de previsibilidad, etc. Para descubrir las limitaciones genuinas de los enfoques ágiles fuera de la revisión de la literatura existente, los autores Agrawal et al. (2016) también realizaron una encuesta en línea donde todavía existen vacíos de investigación en términos de soluciones a los problemas que enfrentan los métodos flexibles. La investigación habla sobre la cultura de desarrollo DevOps para superar muchas de las dificultades del desarrollo de software.

En su investigación sobre DevOps, los autores Mishra & Otaiwi (2020) examinaron los efectos de las características de DevOps en la calidad del sistema. Incrementar la velocidad, la frecuencia de desarrollo y la mejora de calidad del producto son los principales objetivos de DevOps. Como afirman Mishra & Otaiwi (2020), para minimizar las diferencias o llenar todo tipo de brechas de interacción entre los equipos de desarrollo y operaciones, DevOps promete entregar proyectos exitosos con tiempos de implementación o sprint más cortos. Tal como postula Sarlak (2020), para entregar o implementar productos de alta calidad, el autor está de acuerdo en que los equipos de desarrollo y operaciones deben cooperar bien. Aunque para Khan & Shameem (2020) numerosos estudios apoyan el uso de DevOps, otros se oponen y discuten la falta de cuantificación de las medidas de rendimiento y calidad. Los autores Ramzan et al. (2018)

descubrieron que DevOps ayuda a las empresas a lograr sus objetivos y también aumenta su velocidad de reacción a los cambios. En la opinión de Leite et al. (2019), la literatura también confirma la existencia y la implementación exitosa de varios modelos DevOps en la práctica. Tal como expresan Taibi et al. (2019), la cooperación efectiva entre las áreas de Desarrollo y Operaciones conduce a procesos comerciales mejorados que son más adaptables y rápidos de lo habitual, lo que se debe a factores culturales y organizativos que DevOps considera importantes para lograr sus efectos.

Por su parte, Ramos et al. (2017) indican que el desarrollar software involucra todas las actividades requeridas en la creación del producto, las cuales se despliegan en diferentes etapas. En general, estas etapas comprenden la identificación y especificación de los requisitos, la construcción o desarrollo y las pruebas de calidad o validación. Es fundamental que estas actividades o etapas sean transparentes y muy flexibles, con el fin de lograr una gestión adecuada del proyecto.

De acuerdo con Saeedi & Visvizi (2021), el modelo más ampliamente utilizado para desarrollar software es el ciclo de vida del desarrollo de software (SDLC), el cual es una serie de actividades estructuradas. Además, destacan que este ciclo de vida se compone de cinco etapas distintas. La primera etapa es la planificación, donde se lleva a cabo un análisis de riesgos y se determina la viabilidad del proyecto, se estima el costo y los recursos necesarios. La segunda etapa es el análisis, donde se documentan y agrupan todos los requisitos que el producto debe cumplir. La tercera etapa es el diseño, donde se exploran diferentes enfoques para la creación del producto. La cuarta etapa es la fabricación, donde se lleva a cabo la construcción del software de acuerdo con el diseño establecido. Finalmente, la quinta etapa es la prueba, en la cual se verifica la calidad de lo construido y se busca identificar posibles errores, asegurando que se cumplan todos los requisitos establecidos durante la etapa de análisis.

Para Gomes et al. (2022), los procesos de desarrollo que requieren mucho tiempo para la implementación de software son menos adaptables a los clientes en comparación con las implementaciones o lanzamientos frecuentes. El valor del indicador de frecuencia de implementación nos muestra las modificaciones significativas en el software sin encontrarse con obstáculos significativos en términos presupuestarios o de cronograma y nos permite medir el indicador propuesto referente a implementación de software.

Según los autores Narang & Pooja (2022), el indicador definido para evaluar la dimensión calidad del producto, es la estimación de la densidad de defectos. Indican que

la estimación o medida de la cobertura de riesgos informa sobre el porcentaje de cobertura de riesgos en los casos de prueba diseñados. La cobertura del riesgo puede ser mayor con la consideración de riesgos más críticos que probando los triviales. Un valor alto refleja la identificación de más componentes de riesgo y garantiza una entrega de software segura y sin riesgos.

Gallaba (2019) indica que el aumento de la dimensión de productividad del proceso es un beneficio aparente de la implementación continua. Parnin et al. (2017) mencionan que la implementación continua es la práctica de implementar automáticamente el software en el entorno de producción. Para Narang & Mittal (2022) la productividad está directamente asociada con el indicador rendimiento del proceso, el cual indica la cantidad de trabajo realizado en una unidad de intervalo de tiempo. Por lo tanto, se puede decir que la productividad es una medida clara del recuento de lanzamientos y las historias de éxito.

III. METODOLOGÍA

3.1. Tipo y Diseño de Investigación

En base a Ñaupas et al. (2018), las investigaciones aplicadas tienen como objetivo mejorar el funcionamiento de sistemas, procedimientos y normas tecnológicas a partir de los avances científicos y tecnológicos, por lo que su evaluación se basa en criterios de eficiencia, eficacia e ineficacia. De igual manera, Gabriel (2017) menciona que las investigaciones aplicadas buscan aplicar el conocimiento para resolver problemas prácticos. En este sentido, la presente investigación se considera aplicada, ya que se enfoca en la aplicación del marco de trabajo DevOps que busca mejorar la calidad y la efectividad de los procesos para desarrollar software.

3.1.1 Enfoque de Investigación

Sánchez (2019), nos menciona que se puede identificar que una investigación posee enfoque cuantitativo por su habilidad para medir fenómenos utilizando técnicas estadísticas para analizar datos. El principal propósito de este enfoque es describir, explicar, predecir y controlar e identificar los motivos de los acontecimientos de manera imparcial y prever su aparición a través de su detección. Este método se fundamenta en el uso exhaustivo de la medición o cuantificación, tanto en la recopilación de información como en su tratamiento, evaluación de comprensión, por medio de la implementación del procedimiento hipotético-deductivo.

La indagación actual en desarrollo se concentra en la técnica numérica, la cual se trata de una metodología que se fundamenta en la recopilación y examen de información numérica por medio de instrumentos como fichas de registro de observación. Además, se utiliza el método hipotético-deductivo para fundamentar las conclusiones y se realizan pruebas de hipótesis para investigar las respuestas a preguntas concretas relacionadas con el objeto de estudio. Como menciona Rasinger (2013), el enfoque cuantitativo se centra en medir los fenómenos y buscar relaciones estadísticas entre ellos para describir, explicar, predecir y controlar objetivamente sus causas y efectos.

3.1.2 Diseño de Investigación

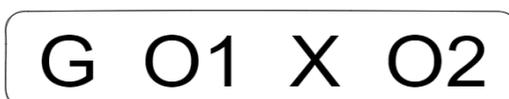
Ñaupas et al. (2018) afirma que el propósito del método experimental es proporcionar orientación adecuada al investigador durante la realización de un experimento, según afirma el autor. El texto también señala que existen diferentes tipos de métodos

experimentales, clasificados como pre-experimentales, cuasi-experimentales y experimentales puros, según su grado de cercanía o similitud con el experimento puro.

Luego Hernández & Mendoza (2018) mencionan que las investigaciones pre-experimentales son diseños con un grupo único. En este contexto, el diseño de pre-test y post-test implica llevar a cabo una evaluación previa al estímulo o tratamiento experimental, seguida de la aplicación del tratamiento y, por último, realizar una prueba posterior al estímulo. En otras palabras, se realiza una evaluación antes y después del experimento. De acuerdo con lo indicado en el párrafo anterior, se considerará que esta investigación será de diseño pre-experimental y con evaluaciones de pre-test y post-test, del cual se tiene un solo grupo, y también no se manipula la variable independiente.

Figura 1

Estudio pre-experimental pre-post



Fuente: Elaboración propia

Dónde:

- G = Grupo pre-experimental
- X = Proceso de desarrollo de software
- O1 = Pre-Test, medición antes de utilizar marco de trabajo DevOps
- O2 = Post-Test, medición después de utilizar marco de trabajo DevOps

3.2. Variables y Operacionalización

Siguiendo las palabras de García (2016), se comprende que una variable se define como una propiedad que respalda al investigador al momento de probar su hipótesis. Por otra parte, los autores Gamarra et al. (2019), definen que las variables se clasifican según el rol que toman en la investigación, distinguiendo entre variables dependientes, independientes, extrañas e intervinientes. Luego, para definir estas variables, se acude a la cita de Arias & Covinos (2021), los cuales describen que la variable en la cual no se interviene (variable independiente) es aquella que se aplica o manipula directamente, con el objetivo de observar los cambios o consecuencias se muestran en la variable que se manipula (variable dependiente). Asimismo, la variable dependiente se encuentra susceptible o sujeta a modificaciones cuando se interviene en la variable independiente.

Según Muhl et al. (2022), nos indican que la definición conceptual de una variable abarca la descripción de sus comportamiento, particularidades y funciones inherentes, pero no explica cómo medirlo. De acuerdo con Laaksonen & Peltoniemi (2018), esta definición implica discernir lo que significa la variable a través del uso de fuentes literarias, como diccionarios, lo cual implica que una variable puede albergar múltiples conceptos.

En base a Romero Urréa et al. (2022), se tiene que la definición operativa para una variable implica la implementación de procedimientos y actividades necesarios para su medición. Siguiendo la perspectiva de Moran & Alvarado (2013), se sostiene que es fundamental identificar la variable más exacta y que abarque lo mayor posible dentro de su contexto, con el propósito de obtener una cantidad mayor de información relevante.

En cuanto a la definición conceptual del Marco de trabajo DevOps tenemos a Faustino et al. (2022), quienes nos señalan que el Desarrollo y Operaciones (DevOps) comprende prácticas organizativas y técnicas que se han implementado en todas las etapas del ciclo para desarrollar software y reducir el tiempo desde el desarrollo del software hasta el entorno de producción, asegurando altos estándares de calidad.

De manera que para la definición operacional respecto al marco de trabajo DevOps, se tomarán aquellas prácticas para lograr un incremento en la calidad, automatización e implementar integración continua y entrega continua para el desarrollo y despliegue de los requerimientos solicitados.

Según la definición conceptual del proceso de desarrollo de software, expuesta por Narang & Mittal (2022), la ingeniería de software se concibe como una disciplina que se enmarca dentro del diseño de software, y se aplica a cada una de las etapas en el desarrollo de software. Estas etapas constan de múltiples fases de desarrollo, que van desde la recopilación de requisitos y un análisis exhaustivo, hasta el diseño, la codificación, las pruebas, la implementación/codificación, las pruebas y el mantenimiento.

Por consiguiente, en la definición operacional con respecto al proceso de desarrollo, esta se medirá mediante fichas de registro. Las dimensiones que se considerarán de la variable dependiente serán: Implementación de software, Calidad del producto y Productividad del proceso. Los cuales se obtendrán de las medidas de los requerimientos solicitados para las mejoras de funcionalidades de software.

3.3 Población, Muestra y Muestreo, Unidad de Análisis

Según Stratton (2021), la definición de población y muestra establece que las unidades de estudio son el conjunto de sujetos que presentan la característica o evento que se estudia y que deben de cumplir con los criterios de aceptación que conforman la población. En el presente caso, la población objetivo se refiere a los requerimientos solicitados de desarrollo de software del departamento de operaciones y tecnología de la entidad financiera, se tomarán los requerimientos que se obtienen en el periodo de tres meses, el cual consta de 112 requerimientos.

En cuanto a Althubaiti (2023), la muestra debe corresponder a una parte representativa del conjunto completo de individuos que comparten similitudes definidas, de la cual se extraen datos relevantes. En consonancia con la definición de Niño (2019) , la muestra se define como una selección de una parte específica de un colectivo o población con el propósito de examinar o evaluar las características que identifican a la totalidad de dicha población.

$$n = \frac{Z^2 pq N}{Ne^2 + Z^2 pq}$$

Del cual se tiene:

Z: Nivel de confianza del 95% (1.96)

E: Error estimado al 5% (0.05)

N: Tamaño de la población de 112 requerimientos

q: $1 - 0.5 = 0.5$

p: 0.5

En el contexto de la presente investigación, aplicando la fórmula, obtenemos que para el tamaño de la muestra es de 87 requerimientos.

Posteriormente, Niño (2019) señala que una muestra aleatoria simple constituye un subconjunto de una población seleccionada de manera aleatoria. En este procedimiento de muestreo, cada integrante de la población posee exactamente la misma probabilidad de ser escogido. El presente estudio es de tipo de aleatorio simple, ya que se cuenta con un tipo de requerimiento.

Sobre la unidad de análisis Ñaupas et al. (2018), señalan que hace referencia a elementos o entidades que comparten propiedades parecidas y se encuentran dentro de un ambiente específico. Desde una perspectiva empírica, estas unidades reflejan las distintivas características, propiedades o atributos inherentes a los objetos, individuos, fenómenos y sucesos a los que se les destinan instrumentos para medir las variables de investigación. Para la presente investigación tenemos como unidad de análisis al requerimiento de desarrollo de software.

Tabla 1

Tabla de unidad de análisis

Elemento	Población	Muestra
<i>Requerimiento</i>	112	87

Fuente: Elaboración propia

3.4 Técnicas e Instrumentos de Recolección de Datos

Según la definición de Chunga (2017), el fichaje se establece como una de las técnicas empleadas por los investigadores para la recopilación y almacenamiento de información. Si queremos llevar a cabo esta técnica, se necesita contar con fichas que faciliten la recopilación y posterior organización de la información extraída tanto de fuentes primarias como secundarias, en concordancia con la naturaleza de la investigación. Según la definición brindada, el fichaje se presenta como una de las técnicas empleadas por los investigadores para recabar y preservar información. Para su aplicación, resulta imprescindible contar con fichas que faciliten la recopilación y posterior organización de los datos extraídos, tanto de fuentes primarias como secundarias, acorde al enfoque particular de la investigación en cuestión. Se utilizará la técnica de fichaje para recopilar la información necesaria de la investigación.

Según la perspectiva de Hernández et al. (2014), un instrumento de recolección de datos se define como aquel que tiene la capacidad de registrar información observable relacionada con los conceptos o variables propuestos por el investigador.

En cuanto a la validez, Niño (2019) se refiere a su capacidad para estimar la variable requerida sin error utilizando las herramientas correctas y apropiadas. Debido a esta característica, el instrumento mide o describe exactamente lo que se pretende medir o describir.

En esta investigación se realiza la evaluación de juicio de expertos que confirman la utilidad del instrumento y la información organizada en él, constatando que los instrumentos son adecuados para su aplicación. Dentro del ámbito de la investigación en cuestión, se empleará la ficha de registro como instrumento para recopilar la información necesaria en todas las dimensiones previamente mencionadas. Cabe destacar que dicha ficha ha sido diseñada específicamente por el propio investigador del presente estudio, con el propósito de recabar información sobre la variable dependiente objeto de estudio y medición.

3.5 Procedimientos

Se aplicaron los instrumentos para la recolección de datos previamente seleccionados y diseñados para este estudio, en este caso, las fichas de registro, a los requerimientos de software. Estas fichas fueron elaboradas incluyendo indicadores que fueron concebidos y validados a través de la evaluación por parte de expertos en la materia. Cabe destacar que se aplicarán en momentos diferentes, se tendrá un antes y un después de la realización de la intervención, estos momentos son conocidos como evaluación de pre-test y post-test.

Los datos requeridos se obtendrán mediante el acceso a la base de información que se tiene de los proyectos, la cual está bajo la propiedad de la entidad financiera correspondiente. Para seleccionar la muestra se considerará como criterio de inclusión a aquellos proyectos que cumplieran con los parámetros y lineamientos del marco de trabajo establecido. Con el fin de garantizar la adecuación de los proyectos, se utilizó una lista de verificación a cada uno de ellos.

Una vez recopilados los datos que se tienen en el pre-test y así como en el post-test, se procederá a realizar el análisis estadístico correspondiente. Este análisis proporcionará la oportunidad de examinar los valores obtenidos de los resultados y validar la eficacia de las mejoras implementadas en el proceso.

3.6 Método de Análisis de Datos

Para realizar el análisis de la data, se usará el análisis cuantitativo de datos, debido a que las variables que se tienen, se pueden expresar en forma de valor numérico. Para obtener las medidas de los datos, se utilizarán métodos estadísticos y luego se pondrán a prueba las hipótesis planteadas. En este análisis, se empleará el software SPSS, que es muy utilizado en el ámbito de la investigación estadística, para llevar a cabo los

cálculos y generar los resultados pertinentes. Para contrastar las hipótesis propuestas, se utilizará la prueba muy conocida de Kolmogórov-Smirnov, ya que este indicador es más sensible a muestras que tienen tamaño igual o mayor a 30 elementos. La cual permitirá comparar los resultados obtenidos en el pre-test con los resultados posteriores a la aplicación del marco de trabajo DevOps en el post-test.

3.7 Aspectos Éticos

De acuerdo con Espinoza (2019), el uso inapropiado de material académico como textos, imágenes y tablas sin la debida referencia es una de las faltas éticas más significativas en la investigación científica, incluso si el investigador está utilizando sus propias publicaciones anteriores. Por lo tanto, es importante ser rigurosos en la gestión de las referencias para prevenir futuros casos que cuestionen a la investigación llevada a cabo.

El informe actual, que pertenece a la indagación realizada, fue creado respetando las normas APA (Asociación Americana de Psicología) versión 7. Además, es importante destacar que el estudio ha sido llevado a cabo siguiendo las directrices establecidas en la RESOLUCIÓN DEL VICERRECTORADO DE INVESTIGACIÓN N°062-2023-VI-UCV, la cual actúa como manual para la elaboración de trabajos que conduzcan a obtener grados y títulos.

También se desea mencionar que se ha utilizado la herramienta Turnitin para evaluar el porcentaje de similitud de este informe con otras publicaciones científicas, de los cuales se han implementado las acciones pertinentes para asegurar que la semejanza se mantenga en niveles mínimos.

IV. RESULTADOS

4.1 Análisis Descriptivo

Indicador 1: Frecuencia de implementación de versiones

Tabla 2

Análisis descriptivo para frecuencia de implementación de versiones

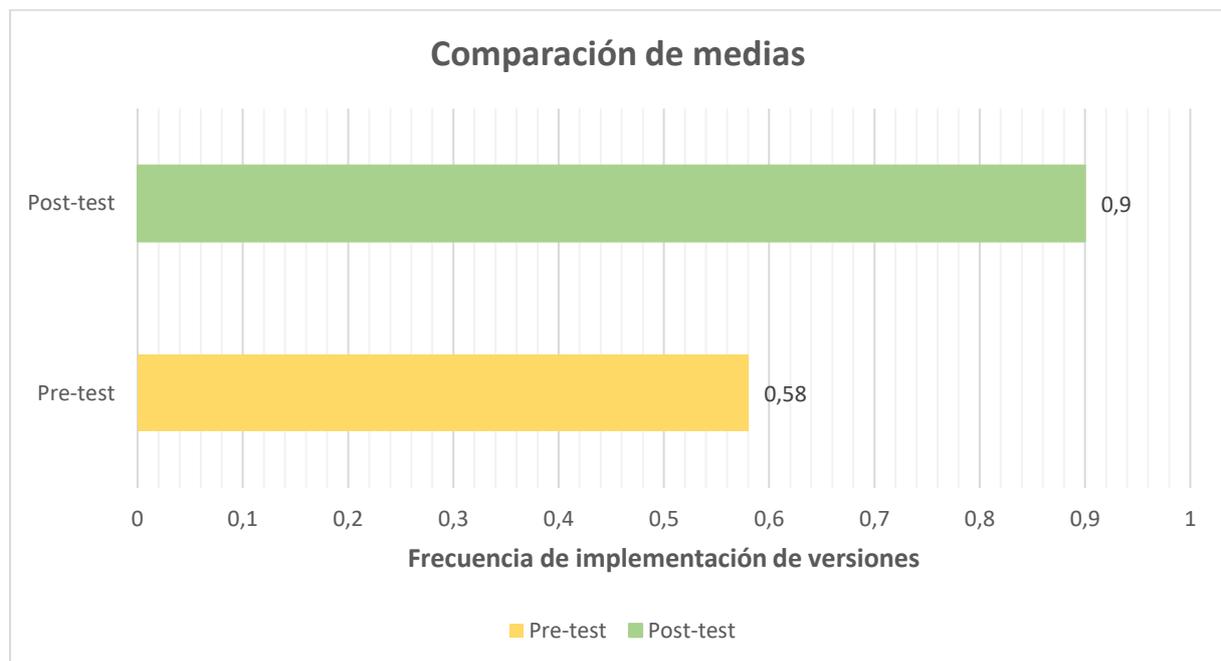
		Muestra	Estadístico
Frecuencia de implementación de versiones	Pre-test	Media	0,5844821
		Mediana	0,476190
		Varianza	0,100
		Desviación estándar	0,31700626
		Mínimo	0,285710
		Máximo	1,666670
	Post-test	Media	0,898012
		Mediana	0,769230
		Varianza	0,148628
		Desviación estándar	0,385523
		Mínimo	0,500000
		Máximo	2,000000

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados que se obtuvieron, se muestran en la tabla 2, referentes al indicador de frecuencia de implementación de versiones, revelaron que el promedio para la muestra del Pre-test fue de 0,58, mientras que el promedio para la muestra del Post-test fue de 0,9. Esto demuestra un incremento en el indicador de frecuencia de implementación de versiones en el grupo sometido al experimento, evidenciando así que la adopción del marco de trabajo DevOps favorece el aumento en la dimensión de implementación de software.

Figura 2

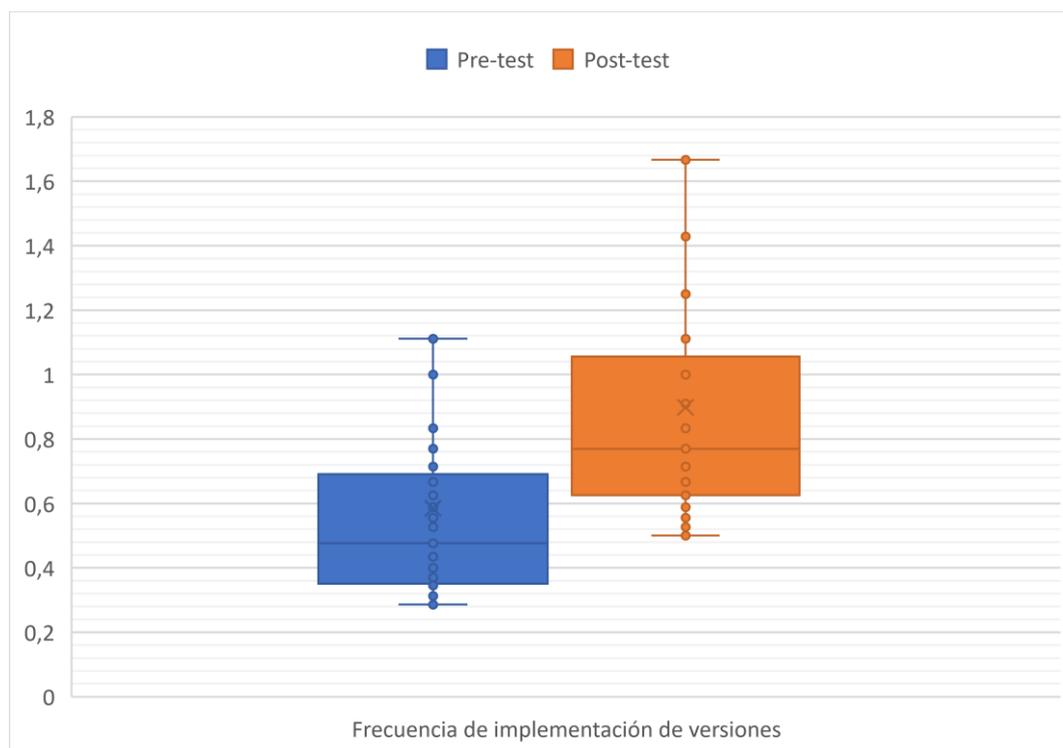
Comparación de las medias para frecuencia de implementación de versiones



Fuente: Elaboración propia

De manera similar, la figura 2 ilustra la variación de la media del indicador de frecuencia de implementación de versiones para ambas muestras. En el grupo del Pre-test, se observa una frecuencia promedio de implementación de versiones de 0,58 por requerimiento. Por otro lado, en el grupo del Post-test, la frecuencia promedio de implementación de versiones por requerimiento es de 0,9. Esto evidencia un incremento significativo de 0,32 implementaciones por requerimiento, lo cual representa un aumento del 55,17%. En conclusión, se puede afirmar que la adopción del marco de trabajo DevOps contribuye al incremento de la implementación de software.

Figura 3
Gráfico de cajas para frecuencia de implementación de versiones



Fuente: Elaboración propia

En el gráfico de cajas presentado en la figura 3, estos datos son referentes al indicador de frecuencia de implementación de versiones, la mediana se muestra para ambos grupos a través de una línea y la media a través de una equis. Mediante este diagrama de cajas, se logró obtener una apreciación visual acerca de la normalidad de los datos. En el caso del Pre-test, el gráfico indica que tanto la mediana como la media son mayores que la mediana ($0,58 > 0,48$), tal como se observó en la tabla 2. Además, se evidencia una mayor concentración de valores por debajo de la media, lo cual sugiere una distribución no normal de los datos. En cuanto al Post-test, el gráfico muestra que la media es mayor que la mediana ($0,9 > 0,77$), de acuerdo a lo reportado en la tabla 2. También se aprecia una mayor concentración de valores por debajo de la media, lo que sugiere una distribución no normal de los datos.

Indicador 2: Densidad de defectos

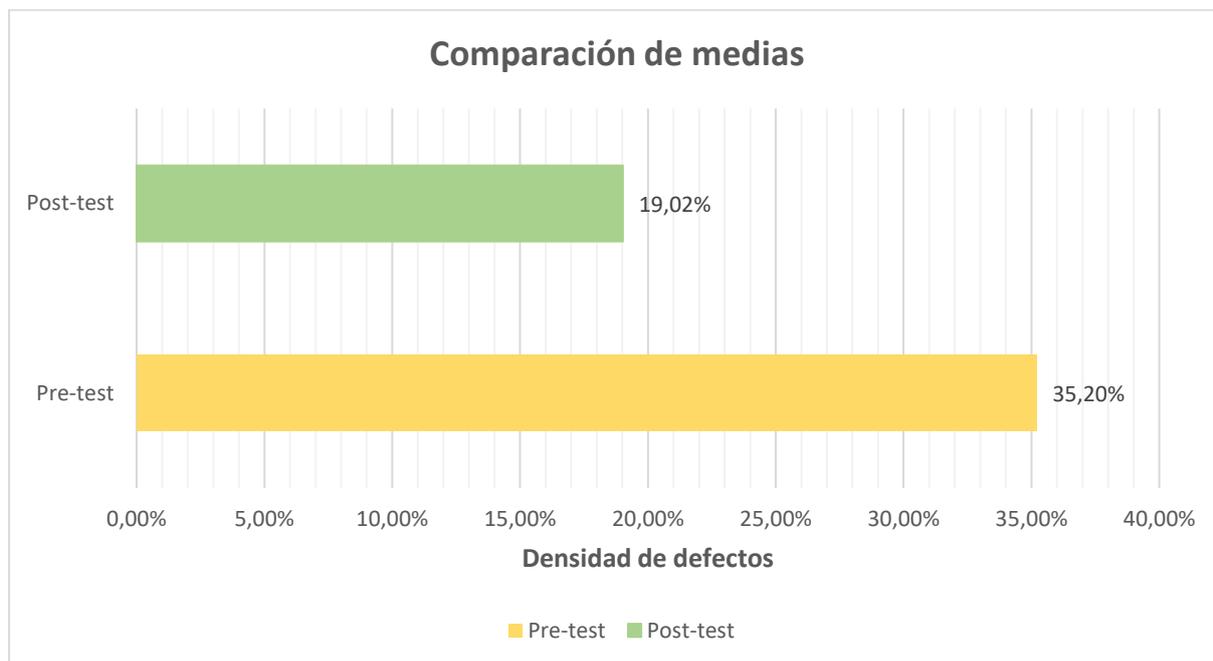
Tabla 3.
Análisis descriptivo para densidad de defectos

Muestra		Estadístico	
Densidad de defectos	Pre-test	Media	0,351985
		Mediana	0,334749
		Varianza	0,003908
		Desviación estándar	0,062516
		Mínimo	0,261354
		Máximo	0,506517
	Post-test	Media	0,190212
		Mediana	0,183575
		Varianza	0,001192
		Desviación estándar	0,034520
		Mínimo	0,141985
		Máximo	0,272729

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados que se obtuvieron, se muestran en la tabla 3, estos datos son referentes al indicador de frecuencia de densidad de defectos, revelaron que el promedio para la muestra del Pre-test fue de 35,2%, mientras que el promedio para la muestra del Post-test fue de 19,02%. Esto demuestra una disminución en el indicador de densidad de defectos en el grupo sometido al experimento, evidenciando así que la adopción del marco de trabajo DevOps contribuye en la mejora de la calidad del producto.

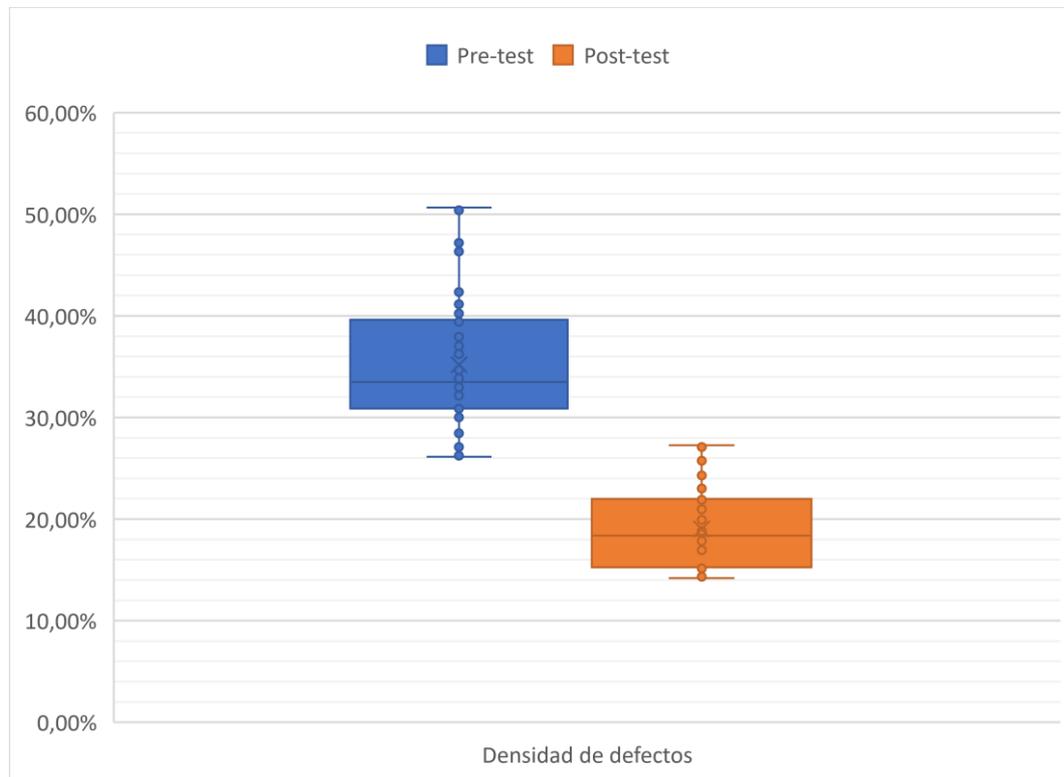
Figura 4
Comparación de las medias para densidad de defectos



Fuente: Elaboración propia

Como se observa en la figura 4, la variación de la media del indicador de densidad de defectos para ambas muestras. En el grupo del Pre-test, se puede apreciar que la densidad de defectos promedio por requerimiento es de 35,2%. Por otro lado, en el grupo del Post-test, la densidad de defectos promedio por requerimiento es de 19,02%. Esto evidencia una reducción significativa del 16,18% en la densidad de defectos para el proyecto. En conclusión, se puede afirmar que la adopción del marco de trabajo DevOps contribuye a mejorar la calidad del producto.

Figura 5
Gráfico de cajas para densidad de defectos



Fuente: Elaboración propia

Del gráfico de cajas mostrado en la figura 5, se observa que los datos son referentes al indicador de densidad de defectos, se utiliza una línea para representar la mediana de ambos grupos y una equis para representar la media. Este gráfico de cajas nos proporcionó una apreciación visual sobre la normalidad de los datos. En el caso del Pre-test, el gráfico indica que la media es mayor que la mediana ($35,2\% > 33,47\%$), tal como se observó en la tabla 3. Además, se observa una mayor concentración de valores por debajo de la media, lo que sugiere una distribución no normal de los datos. En cuanto al Post-test, el gráfico muestra que la media es mayor que la mediana ($19,02\% > 18,36\%$), de acuerdo a lo reportado en la tabla 3. También se aprecia una mayor concentración de valores por debajo de la media, lo que indica una distribución no normal de los datos.

Indicador 3: Productividad del proceso

Tabla 4

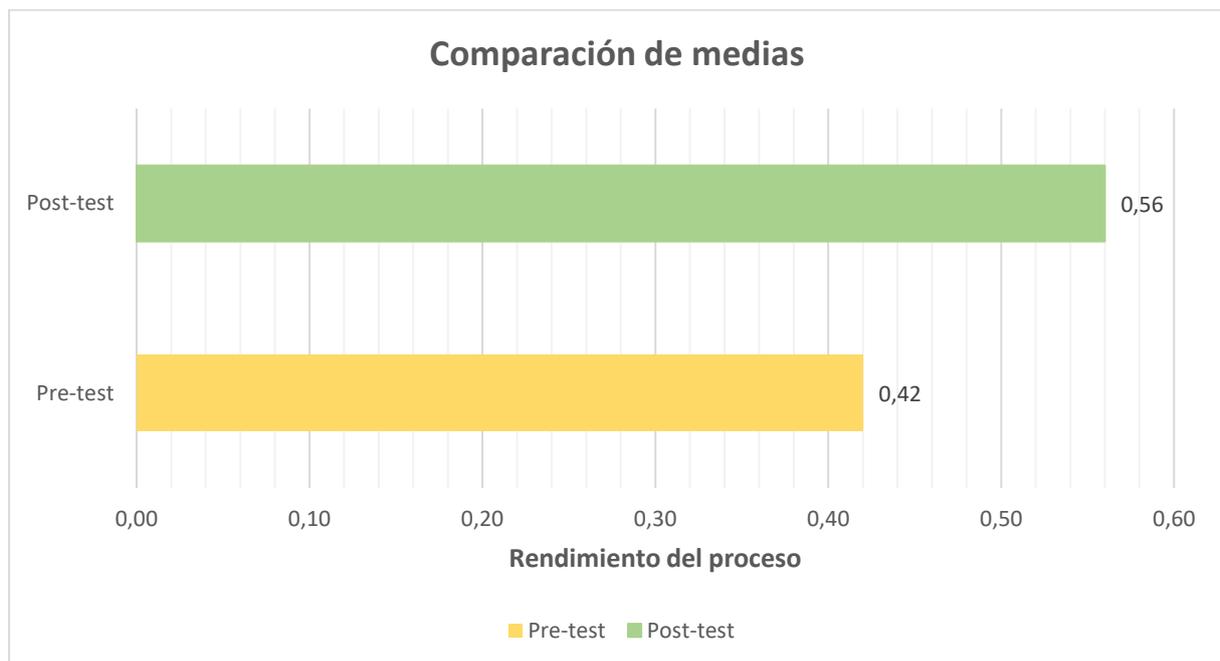
Análisis descriptivo para productividad del proceso

Muestra		Estadístico	
Rendimiento del proceso	Pre-test	Media	0,419923
		Mediana	0,333333
		Varianza	0,075890
		Desviación estándar	0,275482
		Mínimo	0,16667
	Post-test	Máximo	1,00000
		Media	0,557471
		Mediana	0,500000
		Varianza	0,093945
		Desviación estándar	0,306505
	Mínimo	0,200000	
	Máximo	1,000000	

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados que se obtuvieron, se muestran en la tabla 4 revelan que el promedio para la muestra del Pre-test es de 0,42, mientras que el promedio para la muestra del Post-test es de 0,56. Estos datos demuestran un aumento significativo en el indicador de productividad del proceso en el grupo sometido al experimento, evidenciando así que la adopción del marco de trabajo DevOps incrementa el rendimiento del proceso de desarrollo de software.

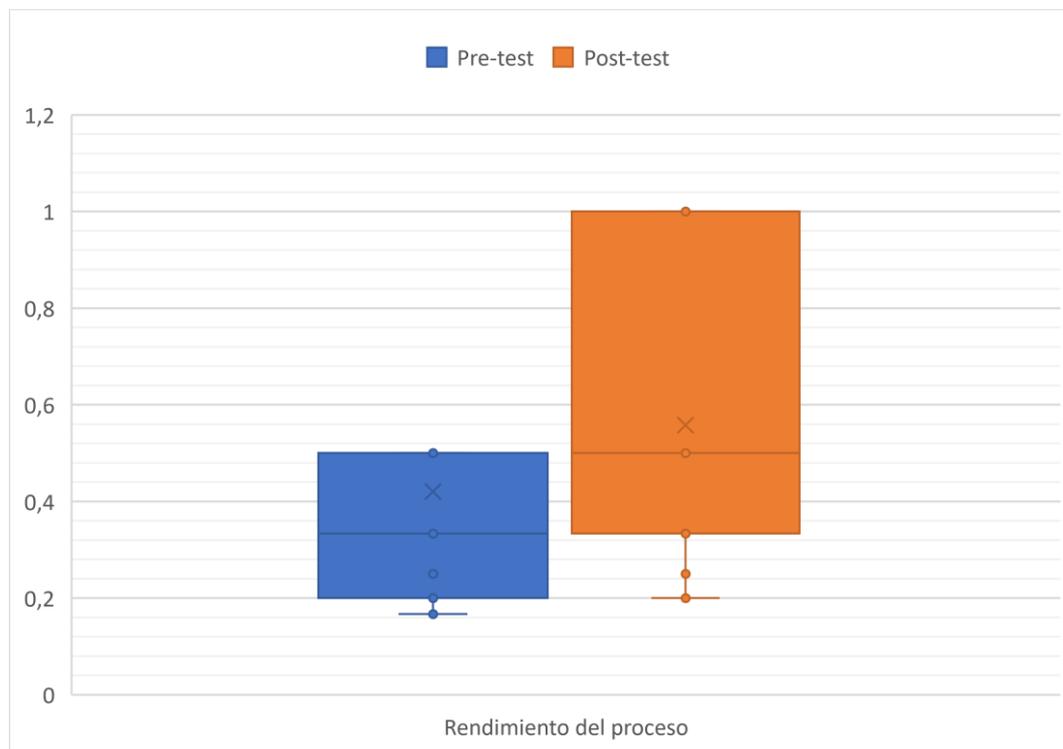
Figura 6
Comparación de las medias para rendimiento del proceso



Fuente: Elaboración propia

En la figura 6, se presenta una comparación de la media del indicador de productividad del proceso de ambos grupos. En el grupo de Pre-test, la productividad del proceso es de 0,42 por requerimiento en promedio. Por otro lado, en el grupo de Post-test, la productividad del proceso por requerimiento es de 0,56 en promedio. Se observa un incremento significativo de 0,14 en la productividad del proceso por requerimiento, lo que representa un aumento del 33,33%. Por consiguiente, se concluye que la adopción del marco de trabajo DevOps resulta en un aumento del rendimiento del proceso.

Figura 7
Gráfico de cajas para densidad de defectos



Fuente: Elaboración propia

En el gráfico de cajas representado en la figura 7, se visualiza la mediana y la media para el indicador de rendimiento del proceso en ambos. Para el grupo de Pre-test, se observa que la media es mayor que la mediana ($0.42 > 0.33$), lo cual concuerda con los datos presentados en la tabla 4. Además, se aprecia una mayor concentración de valores por debajo de la media. Esto nos sugiere que los datos en este grupo no siguen una distribución normal. De manera similar, en el grupo de Post-test, se observa que la media es mayor que la mediana ($0.56 > 0.50$), y se encuentra una mayor concentración de valores por debajo de la media. Esto también nos da a entender que los datos en este grupo siguen una distribución no normal.

4.2 Pruebas de Normalidad

Con el propósito de discernir la presencia de normalidad en los datos, se procedió a utilizar la prueba de Kolmogórov-Smirnov, la cual fue aplicada a cada uno de los indicadores con el fin de determinar si presentaban una distribución normal o no. Posteriormente, se evaluó el nivel de significancia obtenido mediante el valor p y se contrastó con las siguientes hipótesis estadísticas formuladas:

H₀: Los datos presentan una distribución normal

H_a: Los datos no presentan una distribución normal

En el marco de la investigación, se estableció un nivel de confiabilidad del 95%, lo cual conllevó a considerar un margen de error del 5% ($\alpha = 0.05$). Por consiguiente, el valor p fue evaluado siguiendo el siguiente procedimiento:

Si obtenemos el valor de $p < 0.05$, entonces rechazamos la H₀ y aceptamos la H_a.

Si obtenemos el valor de $p \geq 0.05$, entonces aceptamos la H₀ y rechazamos la H_a.

Indicador 1: Frecuencia de implementación de versiones

Tabla 5

Prueba de normalidad para el indicador 1

	Muestra	Kolmogorov-Smirnov		
		Esadístico	gl	Sig. (p)
Frecuencia de implementación de versiones	Pre-test	0.185	87	0.000
	Post-test	0.210	87	0.000

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados de la prueba (tabla 6) pusieron de manifiesto que la significancia del grupo de control (Pre-test) fue inferior a 0.05 ($p = 0.002$), lo que indica la presencia de no normalidad en los datos. De manera similar, se observó que la significancia del grupo experimental también fue menor a 0.05 ($p = 0.001$), lo cual señala que se presenta una distribución no normal. En base a los datos obtenidos, se estableció que si al menos uno de ellos era menor a 0.05 ($p < 0.05$), la hipótesis nula sería descartada. En consecuencia, se concluyó que los datos exhiben una distribución no normal.

Indicador 2: Densidad de Defectos

Tabla 6

Prueba de normalidad para el indicador 2

	Muestra	Kolmogorov-Smirnov		
		Esadístico	gl	Sig. (p)
Densidad de Defectos	Pre-test	0.125	87	0.002
	Post-test	0.129	87	0.001

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados de la prueba (tabla 6) revelaron que la significancia del grupo de control (Pre-test) fue inferior a 0.05 ($p = 0.002$), lo cual indica la presencia de no normalidad en los datos. De manera similar, se observó que la significancia del grupo experimental también fue menor a 0.05 ($p = 0.001$), lo cual indica que los datos presentan una distribución no normal. A partir de los datos obtenidos, se estableció que si al menos uno de ellos era menor a 0.05 ($p < 0.05$), se descartaría la hipótesis nula. En consecuencia, se concluyó que los datos exhiben una distribución no normal.

Indicador 3: Rendimiento del proceso

Tabla 7

Prueba de normalidad para el indicador 2

	Muestra	Kolmogorov-Smirnov		
		Esadístico	gl	Sig. (p)
Rendimiento del proceso	Pre-test	0.290	87	0.000
	Post-test	0.276	87	0.000

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados obtenidos de la prueba (tabla 7) pusieron de manifiesto que la significancia del grupo de control (Pre-test) fue inferior a 0.05 ($p = 0.000$), lo que indicó la presencia de no normalidad en los datos. De manera similar, se observó que la significancia del grupo experimental también fue menor a 0.05 ($p = 0.000$), lo cual indica que los datos presentan una distribución no normal. A partir de los datos obtenidos, se concluyó que si al menos uno de ellos era menor a 0.05 ($p < 0.05$), la hipótesis nula sería

descartada. Por tanto, se llegó a la conclusión de que los datos exhiben una distribución no normal.

4.3 Prueba de hipótesis

Para validar las hipótesis planteadas, se emplearon pruebas estadísticas basadas en la tabla 8, la cual fue obtenida a partir de las pruebas de normalidad.

Tabla 8
Resultados obtenidos de las pruebas de normalidad

Indicador	Normalidad	Pruebas
Frecuencia de implementación de versiones	No normal	U de Mann-Whitney
Densidad de Defectos	No normal	U de Mann-Whitney
Rendimiento del proceso	No normal	U de Mann-Whitney

Fuente: Elaboración propia con aplicativo SPSS v25

Hipótesis específica 1: Frecuencia de implementación de versiones

De acuerdo con los resultados obtenidos, que se exponen en la tabla 8, se procedió a emplear el estadístico no paramétrico de U de Mann-Whitney para validar las siguientes hipótesis:

H0: El marco de trabajo DevOps NO mejora significativamente la implementación de software en el proceso de desarrollo de software en una entidad financiera, Lima 2023.

H1: El marco de trabajo DevOps mejora significativamente la implementación de software en el proceso de desarrollo de software en una entidad financiera, Lima 2023.

Tabla 9
Resultados de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones

Estadístico	Frecuencia de implementación de versiones
U de Mann-Whitney	1580,50
W de Wilcoxon	5408,50
Sig. asintótica (bilateral)	0,000

Fuente: Elaboración propia con aplicativo SPSS v25

Tabla 10

Resumen de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones

Hipótesis nula	Prueba	Sig.	Decisión
El marco de trabajo DevOps NO mejora significativamente la implementación de software en el proceso de desarrollo de software en una entidad financiera, Lima 2023	Prueba U de Mann-Whitney para muestras independientes	0,000	Rechace la hipótesis nula

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados presentados en la tabla 9 corresponden a la aplicación de la prueba no paramétrica, donde se obtuvo un valor de significancia bilateral (Sig. asintótica) de $p = 0.000$. Este valor indica que la hipótesis nula se rechazó (ver tabla 10) y la hipótesis de investigación se aceptó como válida. Por consiguiente, se sustenta que la implementación del marco de trabajo DevOps conlleva un incremento significativo en la implementación de software.

Hipótesis específica 2: Densidad de defectos

De acuerdo con los resultados obtenidos, que se exponen en la tabla 8, se procedió a emplear el estadístico no paramétrico de U de Mann-Whitney para validar las siguientes hipótesis:

H0: El marco de trabajo DevOps NO mejora significativamente la calidad del producto en el proceso de desarrollo de software en una entidad financiera, Lima 2023.

H1: El marco de trabajo DevOps mejora significativamente la calidad del producto en el proceso de desarrollo de software en una entidad financiera, Lima 2023.

Tabla 11

Resultados de la prueba de U-Mann Whitney para el indicador densidad de defectos

Estadístico	Densidad de defectos
U de Mann-Whitney	19,000
W de Wilcoxon	3847,000
Sig. asintótica (bilateral)	0,000

Fuente: Elaboración propia con aplicativo SPSS v25

Tabla 12

Resumen de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones

Hipótesis nula	Prueba	Sig.	Decisión
: El marco de trabajo DevOps NO mejora significativamente la calidad del producto en el proceso de desarrollo de software en una entidad financiera, Lima 2023.	Prueba U de Mann-Whitney para muestras independientes	0.000	Rechace la hipótesis nula

Fuente: Elaboración propia con aplicativo SPSS v25

En la tabla 11 se obtuvieron los resultados que corresponden a la aplicación de la prueba no paramétrica, donde se obtuvo un valor de significancia bilateral (Sig. asintótica) de $p = 0.000$. Esto indica que se rechazó la hipótesis nula (ver tabla 12) y se consideró válida la hipótesis de investigación. Por lo tanto, se pudo sustentar que la implementación del marco de trabajo DevOps incrementa de manera significativa la calidad del producto.

Hipótesis específica 3: Productividad del proceso

De acuerdo con los resultados obtenidos, que se exponen en la tabla 8, se procedió a emplear el estadístico no paramétrico de U de Mann-Whitney para validar las siguientes hipótesis:

H0: El marco de trabajo DevOps NO mejora significativamente el rendimiento en el proceso de desarrollo de software en una entidad financiera, Lima 2023.

H1: El marco de trabajo DevOps mejora significativamente el rendimiento en el proceso de desarrollo de software en una entidad financiera, Lima 2023.

Tabla 13

Resultados de la prueba de U-Mann Whitney para el indicador densidad de defectos

Estadístico	Productividad del proceso
U de Mann-Whitney	2599,000
W de Wilcoxon	6427,000

Sig. asintótica (bilateral)	0,000
-----------------------------	-------

Fuente: Elaboración propia con aplicativo SPSS v25

Tabla 14

Resumen de la prueba de U-Mann Whitney para el indicador frecuencia de implementación de versiones

Hipótesis nula	Prueba	Sig.	Decisión
El marco de trabajo DevOps NO mejora significativamente el rendimiento en el proceso de desarrollo de software en una entidad financiera, Lima 2023.	Prueba U de Mann-Whitney para muestras independientes	0.000	Rechace la hipótesis nula

Fuente: Elaboración propia con aplicativo SPSS v25

Los resultados presentados en la tabla 13 corresponden a la aplicación de la prueba no paramétrica, donde se obtuvo un valor de significancia bilateral (Sig. asintótica) de $p = 0.000$. Esto implica que se rechazó la hipótesis nula (ver tabla 14) y se consideró válida la hipótesis de investigación. Por lo tanto, se sustenta que la implementación del marco de trabajo DevOps incrementa de manera significativa el rendimiento del proceso.

V. DISCUSIÓN

A partir de los datos recopilados en los registros que fueron utilizados para la investigación presente, se infiere de manera general que la implementación de DevOps tiene una influencia de valor positivo para el proceso de desarrollo del software en la entidad financiera de Lima en 2023. Específicamente, se constata que dicha implementación incide favorablemente en los indicadores relacionados con la frecuencia de implementación de versiones, la cobertura de riesgos y la eficiencia del proceso.

Indicador 1: Frecuencia de implementación de versiones.

En relación a la dimensión de implementación de software, medida a través del indicador de frecuencia de implementación de versiones, se pudo observar que, posterior al tratamiento aplicado al grupo experimental, se experimentó un incremento en la frecuencia de implementación de versiones en los sistemas. Los datos recopilados de ambas muestras fueron procesados estadísticamente, revelando medias de 0,58 para el pre-test y 0,90 para el post-test (consultar tabla 5). Esto implica un aumento del 55.17% en la frecuencia de implementación de versiones (figura 3). Por lo tanto, se concluye que la adopción del marco de trabajo DevOps incrementa la frecuencia de implementación de versiones. No obstante, es importante señalar que la mera comparación de las medias no es suficiente para garantizar la existencia de una diferencia y, por ende, una mejora. Para asegurarlo, fue necesario aplicar pruebas de hipótesis en el contexto de las dos muestras independientes.

Para realizar el análisis inferencial, se llevaron a cabo pruebas de normalidad en los datos de cada indicador y se aplicaron pruebas de contraste de hipótesis. En el caso del indicador de frecuencia de implementación de versiones, se evaluó la normalidad en los datos con la prueba de Kolmogorov-Smirnov. Para ambos grupos la significancia fue de 0.00, lo cual nos da a entender que no se tiene una distribución normal. Esto coincide con lo observado en el análisis del gráfico de cajas.

En conclusión, los datos del indicador de frecuencia de implementación de versiones presentan una distribución no normal. Para evaluar la validez de la hipótesis relacionada con este indicador, se utilizó la prueba no paramétrica de U-Mann Withney, dado que se contaba con dos muestras independientes y los datos no seguían una distribución normal. Esta prueba permitió comparar las dos muestras independientes pertenecientes a la misma población y determinar si las medias eran iguales o no. Los

resultados obtenidos (consultar tabla 13) mostraron una significancia (Sig. Asintótica) menor a 0.05 ($p = 0.000$), lo cual no supera el margen de error del 5% establecido. Además, el software estadístico proporcionó un resumen del contraste con una decisión de rechazo (consultar tabla 14), confirmando que se rechazó la hipótesis nula y se aceptó la hipótesis de investigación. Por lo tanto, se concluye que la adopción del marco de trabajo DevOps incrementa de manera significativa la frecuencia de implementación de versiones en el proceso de desarrollo de software.

Los resultados obtenidos en este estudio en cuanto a la implementación de software muestran que en general los requerimientos son atendidos con una mayor frecuencia. Estos resultados son consistentes con los hallazgos reportados por Pastrana-Pardo et al. (2022), quienes aplicaron un modelo de versionamiento, integración y despliegue continuo en el desarrollo de software. El objetivo principal de su estudio fue establecer un entorno de calidad preventiva que implementara controles significativos para el proceso de desarrollo. Además, propusieron la implementación de integración y despliegue continuo (CI/CD). La aplicación de este modelo demostró una reducción del 41.5% en las incidencias reportadas al momento de realizar un despliegue, lo que permitió mejorar tanto la cantidad como la frecuencia de los despliegues realizados. Estos resultados respaldan la idea de que la adopción de prácticas como DevOps puede tener un impacto positivo en la eficiencia y calidad del proceso de desarrollo de software.

Indicador 2: Densidad de defectos

En relación a la dimensión de calidad del producto, medida a través del indicador de densidad de defectos, se pudo observar que, posterior al tratamiento aplicado al grupo experimental, se produjo una reducción en la densidad de defectos en los sistemas. Los datos recopilados de ambas muestras fueron procesados estadísticamente, revelando medias del 35% para el pre-test y del 19% para el post-test (consultar tabla 5). Esto implica una reducción del 16% en la densidad de defectos (figura 3). Por lo tanto, se concluye que la adopción del marco de trabajo DevOps reduce la densidad de defectos. Sin embargo, es importante señalar que la mera comparación de las medias no es suficiente para garantizar la existencia de una diferencia y, por ende, una mejora. Para asegurarlo, fue necesario aplicar pruebas de hipótesis en el contexto de las dos muestras independientes.

Para realizar el análisis inferencial, se llevaron a cabo pruebas de normalidad en los datos de cada indicador y se aplicaron pruebas de contraste de hipótesis. En el caso

del indicador de densidad de defectos, se evaluó la normalidad en los datos con la prueba de Kolmogorov-Smirnov. Para el pre-test la significancia fue de 0.02 y para el post-test fue de 0.01, lo cual nos da a entender que no se tiene una distribución normal. Esto coincide con lo observado en el análisis del gráfico de cajas.

En conclusión, los datos del indicador de densidad de defectos presentan una distribución no normal. Para evaluar la validez de la hipótesis relacionada con este indicador, debido a ello se utilizó la prueba no paramétrica de U-Mann Withney, dado que se contaba con dos muestras independientes y los datos seguían una distribución no normal. Los resultados obtenidos (consultar tabla 13) mostraron una significancia menor a 0.05 ($p = 0.000$), lo cual no supera el margen de error del 5% establecido. Por lo tanto, se concluye que la adopción del marco de trabajo DevOps incrementa de manera significativa la calidad del producto.

Los resultados obtenidos en este estudio en cuanto a la calidad del producto, muestran que en general los requerimientos que son atendidos disminuyen el número de defectos que se tienen para el software. Estos resultados son consistentes con los hallazgos reportados por Marijan et al. (2018), quienes propusieron un enfoque para reducir la duración de los ciclos de prueba en el contexto de la Integración Continua (CI). Al aplicar este enfoque, se logró minimizar el impacto en la efectividad de detección de fallas y en la cobertura de riesgos de las suites de pruebas optimizadas, observando reducciones de hasta un 3%.

Indicador 3: Rendimiento del proceso

En relación a la dimensión de productividad del proceso, medida a través del indicador de rendimiento del proceso, se pudo observar que, posterior al tratamiento aplicado al grupo experimental, se experimentó un incremento en el rendimiento del proceso. Los datos recopilados de ambas muestras fueron procesados estadísticamente, revelando medias de 0,42 para el pre-test y 0,56 para el post-test (consultar tabla 5). Esto implica un aumento del 33.33% en el rendimiento del proceso (figura 3). Por lo tanto, se concluye que la adopción del marco de trabajo DevOps incrementa la productividad del proceso. No obstante, es importante señalar que la mera comparación de las medias no es suficiente para garantizar la existencia de una diferencia y, por ende, una mejora. Para asegurarlo, fue necesario aplicar pruebas de hipótesis en el contexto de las dos muestras independientes.

Para realizar el análisis inferencial, se llevaron a cabo pruebas de normalidad en los datos de cada indicador y se aplicaron pruebas de contraste de hipótesis. En el caso del indicador de rendimiento del proceso, se evaluó la normalidad en los datos con la prueba de Kolmogorov-Smirnov. Para ambos grupos la significancia fue de 0.00, lo cual nos da a entender que no se tiene una distribución normal. Esto coincide con lo observado en el análisis del gráfico de cajas.

Como conclusión, se puede afirmar que los datos correspondientes al indicador de rendimiento del proceso exhiben no normalidad. Se evaluó la validez de la hipótesis mediante la prueba no paramétrica de U-Mann Withney, debido a que se contaba con dos muestras independientes y una distribución no normal de los datos. Los resultados que se muestran en la tabla 13 revelaron una significancia de 0.00 ($p < 0.05$), que no supera el margen de error del 5% establecido. Por lo tanto, se concluye que la adopción del marco de trabajo DevOps incrementa de manera significante el rendimiento del proceso de desarrollo de software.

Los resultados obtenidos en este estudio en cuanto a la productividad del sistema muestran que en el proceso de desarrollo de los requerimientos que son atendidos, muestran una mejora en cuanto al rendimiento de este proceso. Estos resultados son similares a los obtenidos por Ali et al. (2020), quienes realizaron un experimento controlado en el que dos grupos llevaron a cabo un proyecto piloto para validar su propuesta de proceso DevOps híbrido. La implementación de este enfoque permitió lograr una mejora promedio del 35,2% en cuanto al rendimiento del proceso, en comparación con el 10% obtenido sin aplicar el experimento.

VI. CONCLUSIONES

- Primera:** Los resultados del análisis realizado en este estudio han verificado el impacto del marco de trabajo DevOps en el proceso del desarrollo de software en entidades financieras. Se ha observado que su implementación lleva a un aumento en la frecuencia de implementación de software, así como la mejora en calidad de los productos entregados y también en un incremento en la productividad del proceso.
- Segunda:** Este trabajo ha verificado la influencia del Marco de trabajo DevOps en la mejora de la implementación de software, medida a través del indicador de frecuencia de implementación de versiones. Se ha observado un incremento del 55.17% en la frecuencia de implementación de versiones después de implementar el Marco de trabajo DevOps, en comparación con los requerimientos desarrollados en el pre-test. Esto ha llevado a un aumento en la cantidad de despliegues durante el desarrollo de los sistemas en la entidad financiera en el ambiente de producción.
- Tercero:** Este trabajo ha verificado la influencia del Marco de trabajo DevOps en mejorar la calidad del producto, medida a través del indicador de Densidad de defectos. Se ha observado una disminución del 16.18% en la densidad de defectos después de implementar el Marco de trabajo DevOps, en comparación con los requerimientos desarrollados en el pre-test. Con lo cual se disminuyó la densidad de defectos en el desarrollo de los sistemas en la entidad financiera, ya que se disminuyó la cantidad de defectos que se tienen en las fuentes de los aplicativos desarrollados en la entidad financiera.
- Cuarto:** Este trabajo ha verificado la influencia del Marco de trabajo DevOps en mejorar de la productividad del proceso, medida a través del indicador rendimiento del proceso. Se ha observado un incremento del 33.33% luego de implementar el Marco de Trabajo DevOps, en comparación con los requerimientos desarrollados en el pre-test. Con lo cual se incrementó la productividad del proceso durante el desarrollo de los sistemas en la entidad financiera, ya que se incrementaron el rendimiento del proceso de desarrollo de software al incrementar la cantidad de requerimientos atendidos y mejorar el tiempo para completarlos en la entidad financiera.

VII. RECOMENDACIONES

- Primera:** Establecidas las conclusiones de esta investigación se recomienda continuar aplicando el marco de trabajo DevOps, ya que se tiene una mejora en la cantidad de entregas realizadas en producción, se aumenta la calidad del producto y se genera una mayor productividad en el proceso de desarrollo de software.
- Segunda:** Con el fin de continuar con el aumento de frecuencia de implementación de versiones, se recomienda seguir con las mejoras en las configuraciones de los pipelines, que sirven en la automatización de la integración y despliegues del software, lo que conlleva al aumento de lanzamiento a producción de las funcionalidades solicitadas de los requerimientos.
- Tercero:** Con el fin de continuar con la mejora de la calidad del producto, se recomienda seguir con las mejoras en la implementación de herramientas que permitan escanear los proyectos para descubrir defectos en la etapa de desarrollo del producto.
- Cuarto:** Con el fin de continuar con la mejora en la productividad del proceso, se recomienda seguir con la implantación de la cultura DevOps, ya que ello implica tener una mayor colaboración, lo que ayuda a coordinar personas, procesos y herramientas. Mejorando así el rendimiento en el proceso de desarrollo de software.

REFERENCIAS

- Agrawal, A., Atiq, M. A., & Maurya, L. S. (2016). A Current Study on the Limitations of Agile Methods in Industry Using Secure Google Forms. *Procedia Computer Science*, 78, 291–297. <https://doi.org/10.1016/J.PROCS.2016.02.056>
- Ali, N., Daneth, H., & Hong, J. (2020). A hybrid DevOps process supporting software reuse: A pilot project. *Journal of Software: Evolution and Process*, 32(7). <https://doi.org/10.1002/smr.2248>
- Althubaiti, A. (2023). Sample size determination: A practical guide for health researchers. *Journal of General and Family Medicine*, 24(2), 72–78. <https://doi.org/10.1002/jgf2.600>
- Arias, J., & Covinos, M. (2021). *Diseño y metodología de la investigación*. Enfoques Consulting EIRL. <http://hdl.handle.net/20.500.12390/2260>
- Bernal, C. (2016). *Metodología de la Investigación. Administración. Economía. Humanidades y Ciencias Sociales. Cuarta Edición (4th ed.)*. Pearson.
- Casas Fernando. (2020). *Implementación de DevOps para mejorar la integración y despliegue de software en el sector de seguros*. <https://repositorio.ucv.edu.pe/handle/20.500.12692/41593>
- Castro, L., & Espitia, E. (2022). Progress status in the development of software projects supported by Semat and QFD. *INGENIARE - Revista Chilena de Ingeniería*, 30(4), 719–732. <https://eds.s.ebscohost.com/eds/detail/detail?vid=1&sid=e3038418-a0ce-42fe-86df-0f57ea4abb7e%40redis&bdata=JnNpdGU9ZWRzLWxpdmU%3d#AN=161915279&db=fua>
- Chen, M., Wu, Q., & Yang, B. (2019). How Valuable Is FinTech Innovation? *The Review of Financial Studies*, 32(5), 2062–2106. <https://doi.org/10.1093/rfs/hhy130>
- Chugh, M., Chanderwal, N., Upadhyay, R. K., & Punia, D. K. (2021). Antecedents and consequences of knowledge sharing for software process improvement in the Indian software industry. *Journal of Software: Evolution and Process*, 33(2). <https://doi.org/10.1002/smr.2291>

- Chunga, G. (2017). *El estudio y la investigación documental: Estrategias metodológicas y herramientas TIC*.
- Ciric, D., Lalic, B., Gracanin, D., Tasic, N., Delic, M., & Medic, N. (2019). Agile vs. Traditional Approach in Project Management: Strategies, Challenges and Reasons to Introduce Agile. *Procedia Manufacturing*, 39, 1407–1414. <https://doi.org/10.1016/j.promfg.2020.01.314>
- Dhaya Sindhu Battina. (2021). *THE CHALLENGES AND MITIGATION STRATEGIES OF USING DEVOPS DURING SOFTWARE DEVELOPMENT*. www.ijcrt.org
- Dingsøy, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2018). Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1), 490–520. <https://doi.org/10.1007/S10664-017-9524-2/TABLES/6>
- Donnelly, C. (2018). Barclays banks on DevOps to support all-in move to the AWS public cloud. *Computer Weekly*, 6–8.
- Espinoza, D. M. (2019). CONSIDERACIONES ÉTICAS EN EL PROCESO DE UNA PUBLICACIÓN CIENTÍFICA. *Revista Médica Clínica Las Condes*, 30(3), 226–230. <https://doi.org/10.1016/J.RMCLC.2019.04.001>
- Faustino, J., Adriano, D., Amaro, R., Pereira, R., & da Silva, M. M. (2022). DevOps benefits: A systematic literature review. *Software: Practice and Experience*, 52(9), 1905–1926. <https://doi.org/10.1002/spe.3096>
- Gabriel Julio. (2017). *Cómo se genera una investigación científica que luego sea motivo de publicación*. 155–156.
- Gallaba, K. (2019). Improving the Robustness and Efficiency of Continuous Integration and Deployment. *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 619–623. <https://doi.org/10.1109/ICSME.2019.00099>
- Gamarra, G., Wong, F., Pujay, O., & Rivera, T. (2019). *Estadística e Investigación con Aplicaciones de SPSS (2da Edición)*. Editorial San Marcos. http://www.editorialsanmarcos.com/index.php?id_product=226&controller=product&search_query=spss&results=2

- García, J. (2016). *Metodología de la investigación para administradores* (1st ed.). Ediciones de la U. <https://edicionesdelau.com/producto/metodologia-de-la-investigacion-para-administradores/>
- Gomber, P., Kauffman, R. J., Parker, C., & Weber, B. W. (2018). On the Fintech Revolution: Interpreting the Forces of Innovation, Disruption, and Transformation in Financial Services. *Journal of Management Information Systems*, 35(1), 220–265. <https://doi.org/10.1080/07421222.2018.1440766>
- Gomes, M., Pereira, R., Silva, M., de Vasconcelos, J. B., & Rocha, Á. (2022). *KPI's for Evaluation of DevOps Teams* (pp. 142–156). https://doi.org/10.1007/978-3-031-04829-6_13
- Gozman, D., Liebenau, J., & Mangan, J. (2018). The Innovation Mechanisms of Fintech Start-Ups: Insights from SWIFT's Innotribe Competition. *Journal of Management Information Systems*, 35(1), 145–179. <https://doi.org/10.1080/07421222.2018.1440768>
- Hammad, M., & Inayat, I. (2018). Integrating Risk Management in Scrum Framework. *2018 International Conference on Frontiers of Information Technology (FIT)*, 158–163. <https://doi.org/10.1109/FIT.2018.00035>
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (Sexta edición). McGraw-Hill.
- Hobbs, B., & Petit, Y. (2017). Agile Methods on Large Projects in Large Organizations. *Project Management Journal*, 48(3), 3–19. <https://doi.org/10.1177/875697281704800301>
- Khan, A. A., & Shameem, M. (2020). Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process. *Journal of Software: Evolution and Process*, 32(10), e2263. <https://doi.org/10.1002/SMR.2263>
- Laaksonen, O., & Peltoniemi, M. (2018). Operational definitions, variables, and data types in research on dynamic capabilities. *Academy of Management Proceedings*, 2018(1), 13560. <https://doi.org/10.5465/AMBPP.2012.13560abstract>
- Lee, J. C., Hsu, W. C., & Chen, C. Y. (2018). Impact of absorptive capability on software process improvement and firm performance. *Information Technology and Management*, 19(1), 21–35. <https://doi.org/10.1007/s10799-016-0272-6>

- Leite, L., Rocha, C., Kon, F., Milojevic, D., & Meirelles, P. (2019). A Survey of DevOps Concepts and Challenges. *ACM Computing Surveys (CSUR)*, 52(6).
<https://doi.org/10.1145/3359981>
- Lewis, H. (2021). *Ideas That Created the Future* (H. R. Lewis, Ed.). The MIT Press.
<https://doi.org/10.7551/mitpress/12274.001.0001>
- Marijan, D., Liaaen, M., & Sen, S. (2018). DevOps Improvements for Reduced Cycle Times with Integrated Test Optimizations for Continuous Integration. *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 22–27. <https://doi.org/10.1109/COMPSAC.2018.00012>
- Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. *Computer Science Review*, 38, 100308.
<https://doi.org/10.1016/J.COSREV.2020.100308>
- Moran, G., & Alvarado, D. (2013). *Métodos de investigación*. Pearson.
- Muhl, C., Mulligan, K., Bayoumi, I., Ashcroft, R., & Godfrey, C. (2022). *Establishing Internationally Accepted Conceptual and Operational Definitions of Social Prescribing Through Expert Consensus: A Delphi Study*.
- Muñoz, D. A., Ordóñez, H. A., & Buchelli, V. (2021). Lineamientos para la implementación del modelo CALMS de DevOps en mipymes desarrolladoras de software en el contexto surcolombiano. *Revista Guillermo de Ockham*, 18(1), 81–91. <https://doi.org/10.21500/22563202.4270>
- Narang, P., & Mittal, P. (2022). Implementation of DevOps based Hybrid Model for Project Management and Deployment using Jenkins Automation Tool with Plugins. *IJCSNS International Journal of Computer Science and Network Security*, 22(8).
<https://doi.org/10.22937/IJCSNS.2022.22.8.31>
- Narang, P., & Pooja, M. (2022). Performance Assessment of Traditional Software Development Methodologies and DevOps Automation Culture. *Engineering, Technology & Applied Science Research*, 12(6), 9726–9731.
<https://doi.org/https://doi.org/10.48084/etasr.5315>
- Ñaupas, H., Marcelino, P., Valdivia, R., Jesús, D., Palacios, J., Hugo, V., & Delgado, E. R. (2018). *Metodología de la investigación Cuantitativa-Cualitativa y Redacción de la Tesis*.

- Niño, V. (2019). Metodología de la Investigación. Diseño, ejecución e informe. *Ediciones de La U*, 1–165.
- Papazafeiropoulou, A., & Spanaki, K. (2016). Understanding governance, risk and compliance information systems (GRC IS): The experts view. *Information Systems Frontiers*, 18(6), 1251–1263. <https://doi.org/10.1007/s10796-015-9572-3>
- Parnin, C., Helms, E., Atlee, C., Boughton, H., Ghattas, M., Glover, A., Holman, J., Micco, J., Murphy, B., Savor, T., Stumm, M., Whitaker, S., & Williams, L. (2017). The Top 10 Adages in Continuous Deployment. *IEEE Software*, 34(3), 86–95. <https://doi.org/10.1109/MS.2017.86>
- Pastrana-Pardo, M.-A., Ordoñez-Erazo, H.-A., & Cobos-Lozada, C.-A. (2022). Acercamiento a las buenas prácticas para el desarrollo de software basado en DevOps y SCRUM utilizadas en empresas muy pequeñas. *Revista Facultad de Ingeniería*, 31(61), e14828. <https://doi.org/10.19053/01211129.v31.n61.2022.14828>
- Patanakul, P., & Rufo-McCarron, R. (2018). Transitioning to agile software development: Lessons learned from a government-contracted program. *The Journal of High Technology Management Research*, 29(2), 181–192. <https://doi.org/10.1016/j.hitech.2018.10.002>
- Peralta Mori, J. (2019). *DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, Lima 2019*.
- Pilliang, M., & Munawar, M. (2022). Risk Management in Software Development Projects: A Systematic Literature Review. *Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika*, 8(2). <https://doi.org/10.23917/khif.v8i2.17488>
- Ramos, D., Noriega, R., Laínez, J. R., & Durango, A. (2017). *Curso de Ingeniería de Software: 2ª Edición*. <https://books.google.com.pe/books?id=G2Q4DgAAQBAJ&printsec=frontcover#v=onepage&q&f=false>
- Ramzan, M., Shaoib Farooq, M., Zamir, A., Akhtar, W., Ilyas, M., & Ullah Khan, H. (2018). An Analysis of Issues for Adoption of Cloud Computing in Telecom Industries. *Engineering, Technology & Applied Science Research*, 8(4), 3157–3161. <https://doi.org/10.48084/ETASR.2101>

- Rasinger Sebastian. (2013). *Quantitative research in linguistics: An introduction* (A & C Black).
https://books.google.com.pe/books/about/Quantitative_Research_in_Linguistics.html?id=IcyOAQAAQBAJ&redir_esc=y
- Romero Urréa, H., Real Cotto, J. J., Ordoñez Sánchez, J. L., Gavino Díaz, G. E., & Saldarriaga, G. (2022). METODOLOGÍA DE LA INVESTIGACIÓN. *ACVENISPROH Académico*. <https://doi.org/10.47606/ACVEN/ACLIB0017>
- Saeedi, K., & Visvizi, A. (2021). Software Development Methodologies, HEIs, and the Digital Economy. *Education Sciences 2021, Vol. 11, Page 73, 11(2)*, 73.
<https://doi.org/10.3390/EDUCSCI11020073>
- Sánchez Flores, F. A. (2019). Epistemic Fundamentals of Qualitative and Quantitative Research: Consensus and Dissensus Fundamentos epistémicos da pesquisa qualitativa e quantitativa: consensos e dissensos. *Revista Digital de Investigación En Docencia Universitaria*, 101–122. <https://doi.org/10.19083/ridu.2019.644>
- Sankhe, P., Mathur, S., Rehman, T. B., & Dixit, M. (2022). Review of an Agile Software Development Methodology with SCRUM & Extreme Programming. *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, 1–6. <https://doi.org/10.1109/CCET56606.2022.10080640>
- Sarlak, B. (2020). Agile Methodology for Project/Process Management IT System Infrastructure. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–7.
<https://doi.org/10.1109/ICCCNT49239.2020.9225593>
- Stratton, S. J. (2021). Population Research: Convenience Sampling Strategies. *Prehospital and Disaster Medicine*, 36(4), 373–374.
<https://doi.org/10.1017/S1049023X21000649>
- Suárez Barcia, L. (2021). Instrumentos administrativos para el fomento de la innovación tecnológica en el sector financiero peruano. *Derecho PUCP*, 87, 183–229.
<https://doi.org/10.18800/derechopucp.202102.006>
- Suescún-Monsalve, E., Pardo-Calvache, C.-J., Rojas-Muñoz, S.-A., & Velásquez-Uribe, A. (2021). DevOps in Industry 4.0: A Systematic Mapping. *Revista Facultad de Ingeniería*, 30(57), e13314. <https://doi.org/10.19053/01211129.v30.n57.2021.13314>

- Taibi, D., Lenarduzzi, V., & Pahl, C. (2019). Continuous architecting with microservices and DevOps: A systematic mapping study. *Communications in Computer and Information Science*, 1073, 126–151. https://doi.org/10.1007/978-3-030-29193-8_7
- Urresti, S., Ronald, D., Rodríguez, L., & De La Caridad, G. (2019). *SMEs in software development. Software improvement process models in Latin America Contenido*. <https://www.revistaespacios.com/a19v40n28/a19v40n28p09.pdf>
- Zavolokina, L., Dolata, M., & Schwabe, G. (2017). *FinTech Transformation: How IT-Enabled Innovations Shape the Financial Sector* (pp. 75–88). https://doi.org/10.1007/978-3-319-52764-2_6

ANEXOS

Anexo: Matriz de consistencia

Título: Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera, Lima 2023						
Problema	Objetivo	Hipótesis	Variable 1: Marco de trabajo DevOps			
			Variable 2: Proceso de desarrollo de software			
			Dimensiones	Indicadores	Instrumento	Escala
<p>Problemas General: ¿Cómo influye el Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera?</p> <p>Problema específico 1: ¿Cómo influye el Marco de trabajo DevOps en la implementación de software en una entidad financiera?</p> <p>Problema específico 2: ¿Cómo influye el Marco de trabajo DevOps en la calidad del proceso en una entidad financiera?</p> <p>Problema específico 3: ¿Cómo influye el Marco de trabajo DevOps en la productividad del sistema en una entidad financiera?</p>	<p>Objetivo General: Determinar la influencia del Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera</p> <p>Objetivo Específico 1: Determinar la influencia del Marco de trabajo DevOps en la implementación de software en una entidad financiera</p> <p>Objetivo Específico 2: Determinar la influencia del Marco de trabajo DevOps en la calidad del proceso en una entidad financiera</p> <p>Objetivo Específico 3: Determinar la influencia del Marco de trabajo DevOps en la productividad del sistema en una entidad financiera</p>	<p>Hipótesis General: Marco de trabajo DevOps influye significativamente en el proceso de desarrollo de software en una entidad financiera</p> <p>Hipótesis Específica 1: Marco de trabajo DevOps influye significativamente en la implementación de software en una entidad financiera</p> <p>Hipótesis Específica 2: Marco de trabajo DevOps influye significativamente en la calidad del sistema en una entidad financiera</p> <p>Hipótesis Específica 3: Marco de trabajo DevOps influye significativamente en la productividad del sistema en una entidad financiera</p>	<p>Implementación de software</p> <p>Calidad del proceso</p> <p>Rendimiento del proceso</p>	<p>Frecuencia de implementación de versiones (FIV)</p> <p>Densidad de defectos del sistema (DDS)</p> <p>Productividad del proceso (RP)</p>	<p>Ficha de registro</p> <p>Ficha de registro</p> <p>Ficha de registro</p>	<p>Razón</p> <p>Razón</p> <p>Razón</p>
Método y Diseño		Población y muestra		Técnicas e instrumentos		

<p>Enfoque: Cuantitativo</p> <p>Tipo de investigación: Aplicada</p> <p>Diseño de la investigación: Pre experimental</p> <p>De Tipo: Pre-Post con un grupo de control</p>	<p>Población: Los requerimientos solicitados en el desarrollo de software del departamento de operaciones y tecnología de la entidad financiera, en un mes se tienen 112 requerimientos.</p> <p>Tipo de muestreo: Estratificado</p> <p>Tamaño de la muestra: 87 requerimientos</p>	<p>Variable 2: Proceso de desarrollo de software</p> <p>Técnica: Fichaje</p> <p>Instrumento: Ficha de registro</p>
--	--	--

Anexo: Matriz de operacionalización de las variables

Variables de Estudio	Definición Conceptual	Definición Operacional	Dimensión	Indicadores	Formula	Escala
Variable Dependiente: Proceso de desarrollo de software	Narang & Mittal (2022) manifiestan que la ingeniería de software se concibe como una disciplina que se enmarca dentro del diseño de software, y se lleva a cabo a lo largo del ciclo de vida del desarrollo de software (SDLC). Este ciclo consta de múltiples fases de desarrollo, que van desde la recopilación de requisitos y un análisis exhaustivo, hasta el diseño, la codificación, las pruebas, la implementación/codificación, las pruebas y el mantenimiento.	Operacionalmente con respecto al proceso de desarrollo, esta se medirá mediante fichas de registro. Las dimensiones que se considerarán de la variable dependiente serán: Implementación de software, Calidad del proceso y Productividad del sistema. Los cuales se obtendrán sus medidas de los requerimientos solicitados para las mejoras o implementación de funcionalidades de software.	Implementación de software	Frecuencia de implementación de versiones (FIV)	$FIV = \sum_{i=1}^n \frac{N}{T_i}$	Razón
			Calidad del producto	Densidad de defectos del sistema (DDS)	$DDS = \sum_{i=1}^n \frac{Nd_i}{S}$	Razón
			Rendimiento del proceso	Productividad del proceso (RP)	$PP = \sum_{i=1}^n \frac{N}{T_i}$	Razón

Anexo: Instrumentos para la recolección de los datos

Ficha de registro para determinar la implementación de software

	Instrumento: Ficha de registro		Código: FRUCV001	
	Recolección de datos		Página	1 de 1
Investigador	Luis Carlos Moreyra Reyna			
Tipo de prueba	Fichaje			
Variable	Proceso de desarrollo de software			
Indicador	Frecuencia de implementación de versiones			
Fecha de Inicio		Fecha de Fin		
Fórmula	$FIV = \sum_{i=1}^n \frac{N}{T_i}$ <p>FIV: Frecuencia de implementación de versiones</p>			
PRE-TEST				
Ítem	Número de requerimientos	Tiempo de despliegue (hrs.)	Rendimiento del proceso (RP)	
Requerimiento 1				
Requerimiento 2				
Requerimiento 3				
Requerimiento 4				
Requerimiento 5				
Requerimiento n				
POST-TEST				
Ítem	Número de requerimientos	Tiempo de despliegue (hrs.)	Rendimiento del proceso (RP)	
Requerimiento 1				
Requerimiento 2				
Requerimiento 3				
Requerimiento 4				
Requerimiento 5				
Requerimiento n				

Ficha de registro para determinar la calidad del producto

 UNIVERSIDAD CÉSAR VALLEJO	Instrumento: Ficha de registro		Código: FRUCV002	
	Recolección de datos		Página	1 de 1
Investigador	Luis Carlos Moreyra Reyna			
Tipo de prueba	Fichaje			
Variable	Proceso de desarrollo de software			
Indicador	Densidad de defectos del sistema			
Fecha de Inicio		Fecha de Fin		
Fórmula	$DD = \sum_{i=1}^n \frac{Nd_i}{S}$ <p>DD: Densidad de defectos</p>			
PRE-TEST				
Ítem	Número total de defectos	Tamaño del software en KLOC	Densidad de defectos (DD)	
Requerimiento 1				
Requerimiento 2				
Requerimiento 3				
Requerimiento 4				
Requerimiento 5				
Requerimiento n				
POST-TEST				
Ítem	Número total de defectos	Tamaño del software en KLOC	Densidad de defectos (DD)	
Requerimiento 1				
Requerimiento 2				
Requerimiento 3				
Requerimiento 4				
Requerimiento 5				
Requerimiento n				

Ficha de registro para determinar la productividad del proceso

	Instrumento: Ficha de registro		Código: FRUCV003	
	Recolección de datos		Página	1 de 1
Investigador	Luis Carlos Moreyra Reyna			
Tipo de prueba	Fichaje			
Variable	Proceso de desarrollo de software			
Indicador	Rendimiento del proceso			
Fecha de Inicio		Fecha de Fin		
Fórmula	$RP = \sum_{i=1}^n \frac{\text{Número total de requerimientos}}{\text{Tiempo para completarlo (en semanas)}}$ <p>PP: Rendimiento del proceso</p>			
PRE-TEST				
Ítem	Número de requerimientos	Tiempo para completarlo (hrs.)	Rendimiento del proceso (RP)	
Requerimiento 1				
Requerimiento 2				
Requerimiento 3				
Requerimiento 4				
Requerimiento 5				
Requerimiento n				
POST-TEST				
Ítem	Número de requerimientos	Tiempo para completarlo (hrs.)	Rendimiento del proceso (RP)	
Requerimiento 1				
Requerimiento 2				
Requerimiento 3				
Requerimiento 4				
Requerimiento 5				
Requerimiento n				

Anexo: Validación de Instrumentos para la recolección de los datos

Validación de Experto 1:

Evaluación por juicio de expertos

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento "Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera, Lima 2023". La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando al quehacer psicológico. Agradecemos su valiosa colaboración.

1. **Datos generales del juez:**

Nombre del juez:	Adriana Estefani Cáceres Berrospi
Grado profesional:	Maestría (X) Doctor ()
Área de formación académica:	Clínica () Social () Educativa (X) Organizacional ()
Áreas de experiencia profesional:	Tecnologías de información
Institución donde labora:	RENIEC
Tiempo de experiencia profesional en el área:	2 a 4 años () Más de 5 años (X)
Experiencia en Investigación Psicométrica: (si corresponde)	No

2. **Propósito de la evaluación:**

Validar el contenido del instrumento, por juicio de expertos.

3. **Datos de la escala** (Colocar nombre de la escala, cuestionario o inventario)

Nombre de la Prueba:	Ficha de registro de medición
Autor:	Luis Carlos Moreyra Reyna
Procedencia:	La información corresponde al registro de solicitudes de los requerimientos trazados en la planificación conjunta para los equipos de tecnologías de información en una entidad financiera.
Administración:	Unidad de tecnología de información
Tiempo de aplicación:	La información corresponde al registro directo utilizando la técnica del fichaje, de los requerimientos trazados en la planificación conjunta del primer y segundo trimestre del año 2023.
Ámbito de aplicación:	La información recolectada corresponde al registro de solicitudes de los requerimientos trazados en la planificación conjunta para los equipos de tecnologías de información en una entidad financiera, para la toma de decisiones.

Significación:	Explicar Cómo está compuesta la escala (dimensiones, áreas, ítems por área, explicación breve de cuál es el objetivo de medición)
----------------	---

4. Soporte teórico (describir en función al modelo teórico)

Escala/ÁREA	Subescala (dimensiones)	Definición
Ficha de Registro	Implementación de software	La implementación de software en una empresa es el proceso de ejecución y puesta en marcha de un programa que previamente ha sido contratado en la organización. Ahora mismo muchas empresas están acelerando su transformación digital por las demandas del mercado.
	Calidad del producto	La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.
	Rendimiento del proceso	El rendimiento del proceso mide la cantidad de productos terminados que un proceso entrega en un período determinado.

5. Presentación de instrucciones para el juez:

A continuación, a usted le presento el cuestionario elaborado por Luis Carlos Moreyra Reyna en el año 2023 de acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda.

Categoría	Calificación	Indicador
CLARIDAD El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de estas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. totalmente en desacuerdo (nocumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (altonivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.

	4. Alto nivel	El ítem es muy relevante y debe ser incluido.
--	---------------	---

Leer con detenimiento los ítems y calificar en una escala de 1 a 4 su valoración, así como solicitamos brinde sus observaciones que considere pertinente

1. No cumple con el criterio
2. Nivel Bajo
3. Nivel Moderado
4. Alto nivel

Dimensiones del instrumento:

- Primera dimensión: (Implementación de software)**
 Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en la implementación de software del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$FIV = \sum_{i=1}^n \frac{N}{T_i}$$

FIV: Frecuencia de implementación de versiones

N: Número total de despliegues

Ti: Tiempo que toma el despliegue (en horas)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Frecuencia de implementación de versiones	1	4	4	4	

- Segunda dimensión: (Calidad del producto)**
 Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en la calidad del producto del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$DDS = \sum_{i=1}^n \frac{Nd_i}{S}$$

DDS: Densidad de defectos del sistema

Ndi: Número total de defectos

S: Tamaño del software en miles de líneas de código (KLOC)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Densidad de defectos	1	4	4	4	

- **Tercera dimensión:** (Rendimiento del proceso)
Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en el rendimiento del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$PP = \sum_{i=1}^n \frac{N}{T_i}$$

PP: Productividad del proceso

N: Número total de requerimientos completados

Ti: Tiempo transcurrido para completar el requerimiento

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Rendimiento de proceso	1	4	4	4	

Apellidos y nombres del juez validador: Mg. Adriana Estefani Cáceres Berrospi

DNI: 70982067

Lima, 02 de junio de 2023



Firma del experto Informante

Validación de Experto 2:

Evaluación por juicio de expertos

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento "Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera, Lima 2023". La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando al quehacer psicológico. Agradecemos su valiosa colaboración.

1. **Datos generales del juez:**

Nombre del juez:	Nancy Flores Aedo		
Grado profesional:	Maestría (X)	Doctor	()
Área de formación académica:	Clínica ()	Social	()
	Educativa (X)	Organizacional	()
Áreas de experiencia profesional:	Tecnologías de información		
Institución donde labora:	RENIEC		
Tiempo de experiencia profesional en el área:	2 a 4 años ()		
	Más de 5 años (X)		
Experiencia en Investigación Psicométrica: (si corresponde)	No		

2. **Propósito de la evaluación:**

Validar el contenido del instrumento, por juicio de expertos.

3. **Datos de la escala** (Colocar nombre de la escala, cuestionario o inventario)

Nombre de la Prueba:	Ficha de registro de medición
Autor:	Luis Carlos Moreyra Reyna
Procedencia:	La información corresponde al registro de solicitudes de los requerimientos trazados en la planificación conjunta para los equipos de tecnologías de información en una entidad financiera.
Administración:	Unidad de tecnología de información
Tiempo de aplicación:	La información corresponde al registro directo utilizando la técnica del fichaje, de los requerimientos trazados en la planificación conjunta del primer y segundo trimestre del año 2023.
Ámbito de aplicación:	La información recolectada corresponde al registro de solicitudes de los requerimientos trazados en la planificación conjunta para los equipos de tecnologías de información en una entidad financiera, para la toma de decisiones.

Significación:	Explicar Cómo está compuesta la escala (dimensiones, áreas, ítems por área, explicación breve de cuál es el objetivo de medición)
----------------	---

4. Soporte teórico (describir en función al modelo teórico)

Escala/ÁREA	Subescala (dimensiones)	Definición
Ficha de Registro	Implementación de software	La implementación de software en una empresa es el proceso de ejecución y puesta en marcha de un programa que previamente ha sido contratado en la organización. Ahora mismo muchas empresas están acelerando su transformación digital por las demandas del mercado.
	Calidad del producto	La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.
	Rendimiento del proceso	El rendimiento del proceso mide la cantidad de productos terminados que un proceso entrega en un período determinado.

5. Presentación de instrucciones para el juez:

A continuación, a usted le presento el cuestionario elaborado por Luis Carlos Moreyra Reyna en el año 2023 de acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda.

Categoría	Calificación	Indicador
CLARIDAD El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de estas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. totalmente en desacuerdo (no cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.

	4. Alto nivel	El ítem es muy relevante y debe ser incluido.
--	---------------	---

Leer con detenimiento los ítems y calificar en una escala de 1 a 4 su valoración, así como solicitamos brinde sus observaciones que considere pertinente

1. No cumple con el criterio
2. Nivel Bajo
3. Nivel Moderado
4. Alto nivel

Dimensiones del instrumento:

- **Primera dimensión: (Implementación de software)**
Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en la implementación de software del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$FIV = \sum_{i=1}^n \frac{N}{T_i}$$

FIV: Frecuencia de implementación de versiones

N: Número total de despliegues

Ti: Tiempo que toma el despliegue (en horas)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Frecuencia de implementación de versiones	1	4	4	3	

- **Segunda dimensión: (Calidad del producto)**
Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en la calidad del producto del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$DDS = \sum_{i=1}^n \frac{Nd_i}{S}$$

DDS: Densidad de defectos del sistema

Ndi: Número total de defectos

S: Tamaño del software en miles de líneas de código (KLOC)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Densidad de defectos	1	4	4	4	

- **Tercera dimensión:** (Rendimiento del proceso)

Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en el rendimiento del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$PP = \sum_{i=1}^n \frac{N}{T_i}$$

PP: Productividad del proceso

N: Número total de requerimientos completados

Ti: Tiempo transcurrido para completar el requerimiento

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Rendimiento de proceso	1	4	4	4	

Apellidos y nombres del juez validador: Mg. Nancy Flores Aedo

DNI: 41672348

Lima, 02 de junio de 2023



Firma del experto Informante

Validación de Experto 3:

Evaluación por juicio de expertos

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento "Marco de trabajo DevOps en el proceso de desarrollo de software en una entidad financiera, Lima 2023". La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando al quehacer psicológico. Agradecemos su valiosa colaboración.

1. **Datos generales del juez:**

Nombre del juez:	Gisella Rosario Quiñones Bobadilla		
Grado profesional:	Maestría (X)	Doctor	()
Área de formación académica:	Clínica ()	Social	()
	Educativa (X)	Organizacional	()
Áreas de experiencia profesional:	Tecnologías de información		
Institución donde labora:	RENIEC		
Tiempo de experiencia profesional en el área:	2 a 4 años	()	
	Más de 5 años	(X)	
Experiencia en Investigación Psicométrica: (si corresponde)	No		

2. **Propósito de la evaluación:**

Validar el contenido del instrumento, por juicio de expertos.

3. **Datos de la escala** (Colocar nombre de la escala, cuestionario o inventario)

Nombre de la Prueba:	Ficha de registro de medición
Autor:	Luis Carlos Moreyra Reyna
Procedencia:	La información corresponde al registro de solicitudes de los requerimientos trazados en la planificación conjunta para los equipos de tecnologías de información en una entidad financiera.
Administración:	Unidad de tecnología de información
Tiempo de aplicación:	La información corresponde al registro directo utilizando la técnica del fichaje, de los requerimientos trazados en la planificación conjunta del primer y segundo trimestre del año 2023.
Ámbito de aplicación:	La información recolectada corresponde al registro de solicitudes de los requerimientos trazados en la planificación conjunta para los equipos de tecnologías de información en una entidad financiera, para la toma de decisiones.

Significación:	Explicar Cómo está compuesta la escala (dimensiones, áreas, ítems por área, explicación breve de cuál es el objetivo de medición)
----------------	---

4. Soporte teórico (describir en función al modelo teórico)

Escala/ÁREA	Subescala (dimensiones)	Definición
Ficha de Registro	Implementación de software	La implementación de software en una empresa es el proceso de ejecución y puesta en marcha de un programa que previamente ha sido contratado en la organización. Ahora mismo muchas empresas están acelerando su transformación digital por las demandas del mercado.
	Calidad del producto	La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.
	Rendimiento del proceso	El rendimiento del proceso mide la cantidad de productos terminados que un proceso entrega en un período determinado.

5. Presentación de instrucciones para el juez:

A continuación, a usted le presento el cuestionario elaborado por Luis Carlos Moreyra Reyna en el año 2023 de acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda.

Categoría	Calificación	Indicador
CLARIDAD El ítem se comprende fácilmente, es decir, su sintáctica y semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de estas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. totalmente en desacuerdo (no cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencial o importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.

	4. Alto nivel	El ítem es muy relevante y debe ser incluido.
--	---------------	---

Leer con detenimiento los ítems y calificar en una escala de 1 a 4 su valoración, así como solicitamos brinde sus observaciones que considere pertinente

1. No cumple con el criterio
2. Nivel Bajo
3. Nivel Moderado
4. Alto nivel

Dimensiones del instrumento:

- Primera dimensión: (Implementación de software)**
 Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en la implementación de software del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$FIV = \sum_{i=1}^n \frac{N}{T_i}$$

FIV: Frecuencia de implementación de versiones

N: Número total de despliegues

Ti: Tiempo que toma el despliegue (en horas)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Frecuencia de implementación de versiones	1	4	4	3	

- Segunda dimensión: (Calidad del producto)**
 Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en la calidad del producto del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$DDS = \sum_{i=1}^n \frac{Nd_i}{S}$$

DDS: Densidad de defectos del sistema

Ndi: Número total de defectos

S: Tamaño del software en miles de líneas de código (KLOC)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Densidad de defectos	1	4	3	4	

- **Tercera dimensión:** (Rendimiento del proceso)

Objetivos de la Dimensión es determinar la influencia del Marco de trabajo DevOps en el rendimiento del proceso de desarrollo de software en una entidad financiera.

Fórmula:

$$PP = \sum_{i=1}^n \frac{N}{T_i}$$

PP: Productividad del proceso

N: Número total de requerimientos completados

Ti: Tiempo transcurrido para completar el requerimiento

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Rendimiento de proceso	1	4	4	4	

Apellidos y nombres del juez validador: Mg. Gisella Rosario Quiñones Bobadilla

DNI: 20063114

Lima, 02 de junio de 2023



Firma del experto Informante

Anexo: Datos recogidos en las fichas

Indicador 1 – Frecuencia de Implementación de versiones (PRE)

	# Total de requerimientos	Tiempo que toma el despliegue en horas
R001	1	2,2
R002	1	0,6
R003	1	1,8
R004	1	2,6
R005	1	3,1
R006	1	3,1
R007	1	2,4
R008	1	2,5
R009	1	2,7
R010	1	3,4
R011	1	3,3
R012	1	1,4
R013	1	1,9
R014	1	1,3
R015	1	0,8
R016	1	2,5
R017	1	0,7
R018	1	2,8
R019	1	0,9
R020	1	2,2
R021	1	3,5
R022	1	2,2
R023	1	2,4
R024	1	1
R025	1	0,8
R026	1	2,9
R027	1	1,5
R028	1	2,4
R029	1	3,1
R030	1	1,3
R031	1	1,6
R032	1	0,9
R033	1	1,5
R034	1	1,2
R035	1	1,8
R036	1	2,7
R037	1	2
R038	1	3,2
R039	1	2,6
R040	1	2

R041	1	1,3
R042	1	2,5
R043	1	1,9
R044	1	0,8
R045	1	2,1
R046	1	2,4
R047	1	3
R048	1	3
R049	1	1,7
R050	1	1,3
R051	1	0,9
R052	1	1,4
R053	1	3,5
R054	1	3,2
R055	1	1,2
R056	1	3,2
R057	1	0,9
R058	1	1,7
R059	1	2
R060	1	0,7
R061	1	3,5
R062	1	2
R063	1	2,9
R064	1	1,6
R065	1	1,7
R066	1	2,9
R067	1	2,3
R068	1	3,2
R069	1	1,3
R070	1	2,4
R071	1	1,9
R072	1	3,5
R073	1	0,9
R074	1	2,9
R075	1	1,9
R076	1	1,9
R077	1	3,1
R078	1	2,1
R079	1	2,8
R080	1	1,8
R081	1	2,7
R082	1	0,8
R083	1	3,4
R084	1	2,6
R085	1	3,4

R086	1	1,7
R087	1	1,5

Indicador 1 – Frecuencia de Implementación de versiones (POST)

	# Total de requerimientos	Tiempo que toma el despliegue en horas
R001	1	1
R002	1	1,9
R003	1	1,6
R004	1	0,7
R005	1	1,3
R006	1	1,1
R007	1	1,4
R008	1	1,4
R009	1	1,2
R010	1	0,6
R011	1	0,9
R012	1	1,1
R013	1	0,5
R014	1	1
R015	1	1,7
R016	1	1,6
R017	1	1,7
R018	1	0,8
R019	1	0,9
R020	1	0,7
R021	1	1,7
R022	1	1,9
R023	1	0,9
R024	1	1,6
R025	1	1,3
R026	1	0,9
R027	1	1,8
R028	1	0,7
R029	1	1,3
R030	1	1,5
R031	1	1,1
R032	1	1
R033	1	1,4
R034	1	0,5
R035	1	1,5
R036	1	0,5
R037	1	0,9
R038	1	2

R039	1	1,1
R040	1	1,4
R041	1	1,3
R042	1	1,9
R043	1	0,9
R044	1	1,6
R045	1	1,8
R046	1	1,2
R047	1	0,7
R048	1	0,5
R049	1	1,2
R050	1	1,3
R051	1	1,9
R052	1	1,4
R053	1	0,6
R054	1	1,2
R055	1	1,7
R056	1	1,5
R057	1	1,2
R058	1	1,8
R059	1	1,3
R060	1	1,5
R061	1	1
R062	1	1,1
R063	1	1,3
R064	1	1,9
R065	1	1,6
R066	1	1,7
R067	1	1,3
R068	1	1,6
R069	1	1,2
R070	1	1,6
R071	1	1,4
R072	1	1,9
R073	1	1,3
R074	1	2
R075	1	1,9
R076	1	0,6
R077	1	1,9
R078	1	1,4
R079	1	0,6
R080	1	1,3
R081	1	0,9
R082	1	1,2
R083	1	0,7

R084	1	1,4
R085	1	1,3
R086	1	1,6
R087	1	0,8

Indicador 2 – Densidad de Defectos (PRE)

	# Total de defectos	# Miles Líneas de código
R001	18	55765
R002	18	55957
R003	19	56141
R004	21	56320
R005	17	56648
R006	19	56759
R007	19	57126
R008	19	57352
R009	22	57566
R010	22	57742
R011	21	57929
R012	22	57999
R013	19	58324
R014	16	58713
R015	16	58747
R016	22	58876
R017	25	59041
R018	30	59228
R019	30	59537
R020	28	59633
R021	20	59714
R022	20	60101
R023	20	60173
R024	20	60282
R025	28	60423
R026	22	60584
R027	20	60673
R028	24	60871
R029	29	61107
R030	29	61314
R031	29	61485
R032	29	61713
R033	29	61745
R034	29	61927
R035	25	62130
R036	25	62250
R037	25	62478

R038	25	62773
R039	26	63008
R040	26	63189
R041	26	63530
R042	26	63579
R043	26	63856
R044	26	64145
R045	26	64511
R046	20	64704
R047	20	64815
R048	20	65006
R049	19	65399
R050	19	65528
R051	19	65798
R052	19	65985
R053	19	66353
R054	19	66394
R055	19	66569
R056	19	66819
R057	22	67179
R058	22	67488
R059	22	67870
R060	22	68115
R061	22	68107
R062	18	68373
R063	18	68581
R064	18	68872
R065	23	69222
R066	23	69444
R067	23	69525
R068	23	69894
R069	19	69935
R070	19	70093
R071	19	70458
R072	19	70737
R073	19	70795
R074	19	70982
R075	25	71236
R076	25	71485
R077	25	71860
R078	25	72107
R079	25	72419
R080	25	72461
R081	27	72551
R082	27	72638

R083	27	72761
R084	27	72814
R085	27	72896
R086	23	73088
R087	23	73333

Indicador 2 – Densidad de Defectos (POST)

	# Total de defectos	# Miles Líneas de código
R001	20	73333
R002	20	73781
R003	19	73787
R004	18	74082
R005	14	74434
R006	11	74805
R007	11	74944
R008	11	75254
R009	16	75640
R010	19	75944
R011	18	76097
R012	19	76125
R013	16	76381
R014	13	76697
R015	13	76778
R016	19	76844
R017	14	76962
R018	14	77333
R019	14	77598
R020	14	77716
R021	14	78075
R022	14	78160
R023	14	78215
R024	14	78385
R025	12	78609
R026	12	78672
R027	12	78835
R028	12	78933
R029	12	79225
R030	12	79278
R031	12	79481
R032	12	79840
R033	12	79860
R034	12	79987
R035	17	80050

R036	17	80114
R037	17	80252
R038	17	80362
R039	18	80741
R040	18	81135
R041	18	81454
R042	18	81727
R043	18	82104
R044	18	82168
R045	18	82207
R046	17	82584
R047	17	82785
R048	17	83074
R049	12	83310
R050	12	83609
R051	12	83838
R052	16	83960
R053	16	83978
R054	16	84248
R055	12	84516
R056	16	84554
R057	19	84869
R058	19	85039
R059	19	85417
R060	19	85730
R061	19	85933
R062	15	86017
R063	15	86102
R064	15	86254
R065	20	86596
R066	20	86734
R067	20	86970
R068	20	86978
R069	16	86984
R070	16	87158
R071	16	87291
R072	16	87521
R073	16	87870
R074	16	88047
R075	13	88359
R076	13	88464
R077	13	88750
R078	13	88950
R079	13	89143
R080	13	89462

R081	18	89798
R082	18	89900
R083	14	89910
R084	14	90207
R085	18	90383
R086	16	90596
R087	20	90791

Indicador 3 – Rendimiento del proceso (PRE)

	# total de requerimientos completados	Tiempo transcurrido para completarlo
R001	1	2
R002	1	1
R003	1	5
R004	1	4
R005	1	5
R006	1	5
R007	1	4
R008	1	4
R009	1	3
R010	1	1
R011	1	3
R012	1	3
R013	1	3
R014	1	5
R015	1	2
R016	1	1
R017	1	1
R018	1	5
R019	1	3
R020	1	3
R021	1	5
R022	1	4
R023	1	5
R024	1	3
R025	1	4
R026	1	5
R027	1	3
R028	1	3
R029	1	3
R030	1	5
R031	1	1
R032	1	3

R033	1	3
R034	1	5
R035	1	5
R036	1	2
R037	1	5
R038	1	3
R039	1	1
R040	1	4
R041	1	4
R042	1	5
R043	1	3
R044	1	2
R045	1	5
R046	1	2
R047	1	6
R048	1	2
R049	1	3
R050	1	1
R051	1	2
R052	1	3
R053	1	5
R054	1	3
R055	1	1
R056	1	3
R057	1	1
R058	1	3
R059	1	3
R060	1	5
R061	1	2
R062	1	4
R063	1	4
R064	1	4
R065	1	2
R066	1	5
R067	1	5
R068	1	5
R069	1	4
R070	1	1
R071	1	2
R072	1	6
R073	1	2
R074	1	5
R075	1	2
R076	1	2
R077	1	1

R078	1	1
R079	1	2
R080	1	5
R081	1	1
R082	1	6
R083	1	3
R084	1	1
R085	1	2
R086	1	4
R087	1	3

Indicador 3 – Rendimiento del proceso (POST)

	# total de requerimientos completados	Tiempo transcurrido para completarlo
R001	1	3
R002	1	2
R003	1	1
R004	1	1
R005	1	5
R006	1	1
R007	1	3
R008	1	5
R009	1	3
R010	1	5
R011	1	3
R012	1	2
R013	1	5
R014	1	1
R015	1	2
R016	1	4
R017	1	3
R018	1	3
R019	1	3
R020	1	1
R021	1	4
R022	1	4
R023	1	2
R024	1	5
R025	1	1
R026	1	4
R027	1	2
R028	1	4
R029	1	4
R030	1	2
R031	1	1

R032	1	3
R033	1	4
R034	1	3
R035	1	4
R036	1	2
R037	1	1
R038	1	4
R039	1	2
R040	1	3
R041	1	2
R042	1	1
R043	1	1
R044	1	2
R045	1	2
R046	1	2
R047	1	1
R048	1	1
R049	1	1
R050	1	1
R051	1	2
R052	1	3
R053	1	1
R054	1	2
R055	1	4
R056	1	1
R057	1	2
R058	1	2
R059	1	2
R060	1	3
R061	1	2
R062	1	1
R063	1	1
R064	1	4
R065	1	1
R066	1	2
R067	1	2
R068	1	2
R069	1	1
R070	1	4
R071	1	3
R072	1	1
R073	1	1
R074	1	2
R075	1	3
R076	1	4

R077	1	4
R078	1	1
R079	1	2
R080	1	2
R081	1	1
R082	1	4
R083	1	3
R084	1	1
R085	1	1
R086	1	4
R087	1	2

Anexo: Metodología

1. Introducción

Esta investigación tiene como práctica, describir el cómo implementar y configurar una serie de herramientas que conformen un nuevo modelo, idea o marco de trabajo como lo es DevOps, y será llevado a cabo dentro del Área de Tecnología de la entidad financiera.

DevOps (Development and Operations) surgió a finales de la pasada década como una solución necesaria a adoptar para un mejor funcionamiento del desarrollo ágil, el cual se basa en un desarrollo iterativo e incremental donde las entregas de paquetes de código son cada vez de mayor frecuencia (Fig. 1).

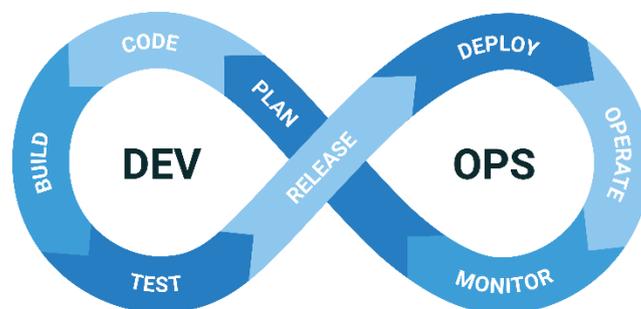


Fig. 1 Flujo de tareas de DevOps (octopus.com)

Lo que propone DevOps es establecer una amplia comunicación, colaboración e integración entre los desarrolladores que forman el departamento IT y los profesionales que forman el departamento de operaciones, ya que estos elementos son clave para que el sistema funcione y como consecuencia se vea reflejado en beneficios para la empresa. Para ello, se han de implementar una serie de herramientas que faciliten esta metodología de trabajo:

- Administración del código.
- Integración continua para la compilación, ejecución de pruebas y detección de errores del código.
- Repositorios de artefactos donde almacenar el código durante el desarrollo.
- Gestión de cambios
- Configuración de la infraestructura.
- Monitoreo del rendimiento de la aplicación.

Si bien es cierto que no existe una implementación determinada ni tampoco el uso de un software en el mercado en concreto para cada una de las herramientas mencionadas. En el siguiente apartado se analizará cuáles son las diferentes alternativas que podemos encontrar.

Por último, cabe decir que en la empresa existen ciertos proyectos en los que ya se implementa la metodología DevOps, pero que tienen características que difieren con las de esta investigación como por ejemplo el tipo de herramienta de control de versiones. De igual manera, se tendrán en cuenta a la hora de poder adaptarlas a nuestras necesidades.

2. Objetivos

El objetivo principal de la investigación se centra en implementar y configurar una serie de herramientas que permitan aplicar la metodología DevOps para el proceso de desarrollo de software en la entidad financiera.

Es muy común que durante el ciclo de vida del software surjan problemas e inconvenientes, que dependan directa o indirectamente de nosotros, o que simplemente no se puedan predecir al inicio, por esta razón y para que el proyecto tenga una base y cumpla unos mínimos, los objetivos se categorizan en tres clases: prioritarios y generales.

2.1 Prioritarios

- El diseño de la arquitectura ha de ser de fácil mantenimiento.
- El diseño de la arquitectura ha de ser fácilmente ampliable.
- Tener un sistema de control de versiones integrado en el sistema.
- Configuración y administración de Azure Devops como sistema de automatización.
 - Chequeo continuo de modificaciones del repositorio principal.
 - Automatización de la compilación del código fuente.
 - Automatización del test del software.
 - Automatización de la entrega y despliegue del software.

2.2 Generales

- Configuración y administración de Azure Devops como sistema de automatización.
 - Monitoreo de los pasos en la construcción del software.

- Notificación a los desarrolladores de los posibles errores.
- Configuración de los servicios en Azure como ACR y AKS para el despliegue.

3. Metodología

Como esta investigación está orientada a promover el desarrollo ágil, a la hora de decidir una metodología de trabajo también se tuvo en cuenta trabajar con metodologías ágiles que se usan actualmente en la entidad financiera, ya que permiten gestionar nuestro proyecto de forma flexible y autónoma reduciendo costes e incrementando nuestra productividad. Dentro de la que se utiliza en la mayoría de los equipos tenemos a Scrum.

La herramienta Jira ha permitido tener una visión general del flujo de trabajo y el estado en que se encontraban los requerimientos en cada momento (Fig. 2a y 2b).

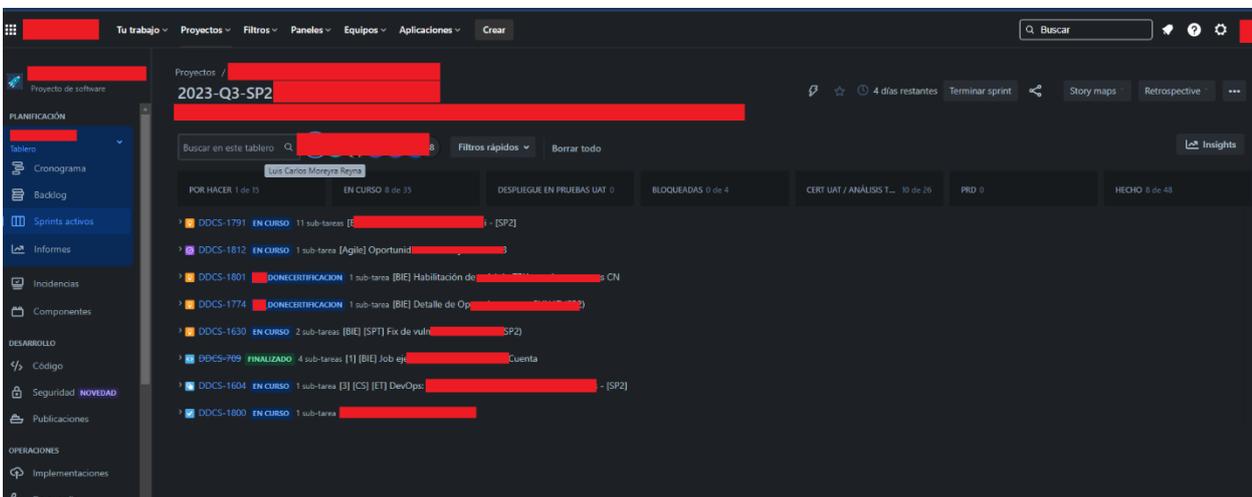


Fig. 2a Tablero de requerimientos en JIRA

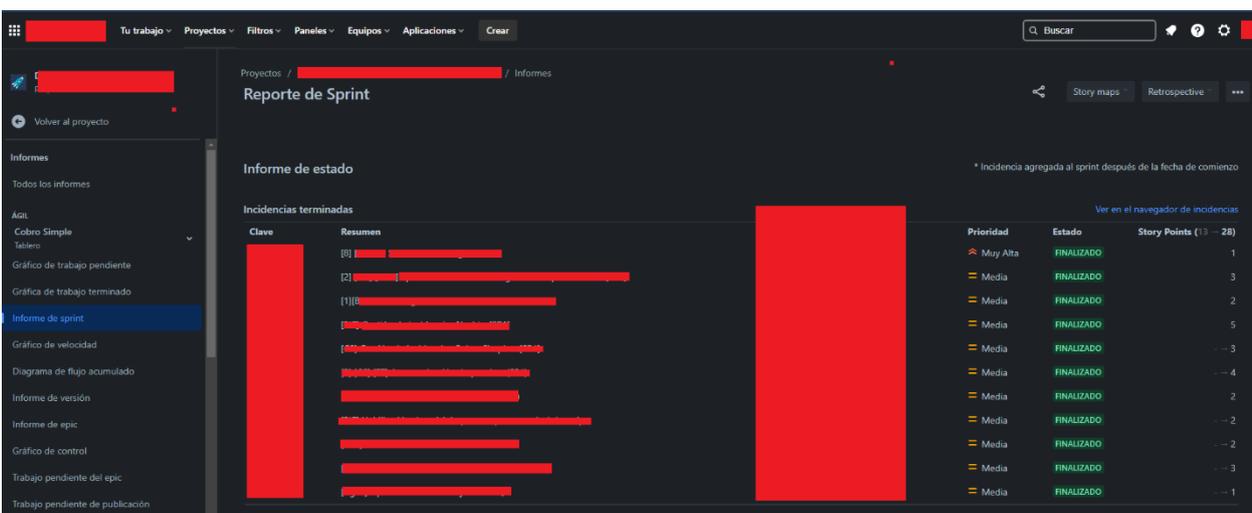


Fig. 2b Histórico de requerimientos en JIRA

3.1 Planificación

Para llevar un control a medida en el avance y poder conseguir los objetivos de esta investigación, se elaboró una planificación. Esta planificación se dividió en cinco fases con un conjunto de actividades a realizar en cada una:

- **Fase inicial.** Se definieron las necesidades y requisitos, se marcaron los objetivos, se elaboró una planificación con unas tareas a realizar.
- **Diseño.** Se especificaron las herramientas a utilizar para implementar las funcionalidades del sistema y se elaboró un diagrama general de la infraestructura del sistema.
- **Desarrollo.** Se llevaron a cabo las implementaciones y configuraciones de cada uno de los componentes y funcionalidades del sistema, así como la integración de todos los componentes en la infraestructura.
- **Pruebas.** Se realizaron las pruebas que validarán los requisitos, se analizaron los resultados y variaciones respecto a las estimaciones esperadas.
- **Cierre.** Se aseguró que todas las actividades de la investigación hayan quedado correctamente finalizadas, se realizó una evaluación general de las tareas implementadas.

4. Requisitos

A continuación, se enumerarán los requisitos funcionales y no funcionales que determinarán el comportamiento de nuestra implementación. A la hora de recoger los requisitos se ha tenido en cuenta el “qué” y el “cómo” se realizará el trabajo en general, es decir, el conjunto de herramientas que forman nuestra implementación de DevOps.

4.1 Funcionales

La siguiente lista recoge los requisitos funcionales:

- RF1. El sistema ha de permitir trabajar en equipo.
- RF2. El sistema ha de revisar el repositorio central de manera continua.
- RF3. El sistema ha de automatizar la compilación del código fuente.
- RF4. El sistema ha de automatizar los tests del código fuente.
- RF5. El sistema ha de automatizar la entrega (empaquetado) del código fuente.
- RF6. El sistema ha de automatizar el despliegue del código fuente a los servicios de Azure.

- RF7. El sistema ha de notificar a los desarrolladores del resultado de la ejecución de la cadena de procesos.
- RF8. Los contenedores han de poder establecer conexión SSH.

4.2 No funcionales

La siguiente lista recoge los requisitos no funcionales:

- RNF1. El proyecto en desarrollo ha de ser del tipo jar utilizando el framework Spring boot.
- RNF2. La herramienta que almacenará los repositorios centrales será Bitbucket.
- RNF3. El sistema de control de versiones ha de ser Git.
- RNF4. El sistema que automatizará la cadena de procesos ha de ser la herramienta Azure DevOps.
- RNF5. La ejecución de la cadena de procesos de Azure DevOps no ha de ser superior a 5 minutos.

5. Diseño

En la fase de diseño, una vez estudiado los lineamientos de la empresa y evaluado las alternativas para determinar cuáles serían las herramientas por usar, implementar y configurar, se procedió a diseñar el aspecto que tendría el pipeline de CI/CD (Fig. 3).

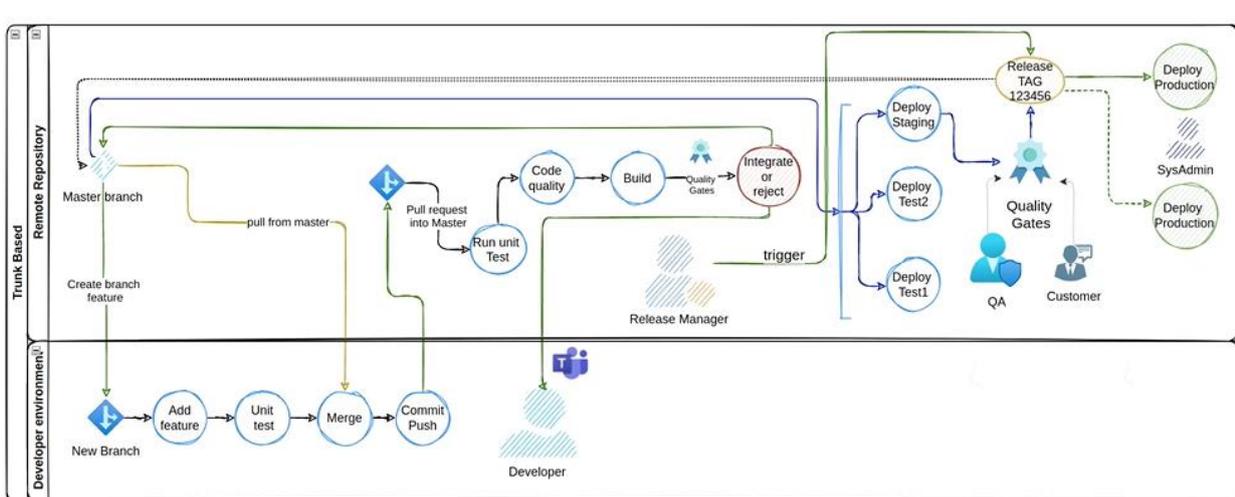


Fig. 3 Pipeline de integración y despliegue continuo

5.1 Azure

Utilizar Azure en esta investigación fue una idea que se fijó siguiendo los lineamientos por parte del área de arquitectura de la entidad financiera. Una de las motivaciones que se tuvo en cuenta en este proyecto fue tener la posibilidad de dedicar horas tanto en horario de oficina como fuera. El hecho de tener los proyectos en la nube posibilita acceder a él desde cualquier lugar con acceso a internet (Fig. 4).

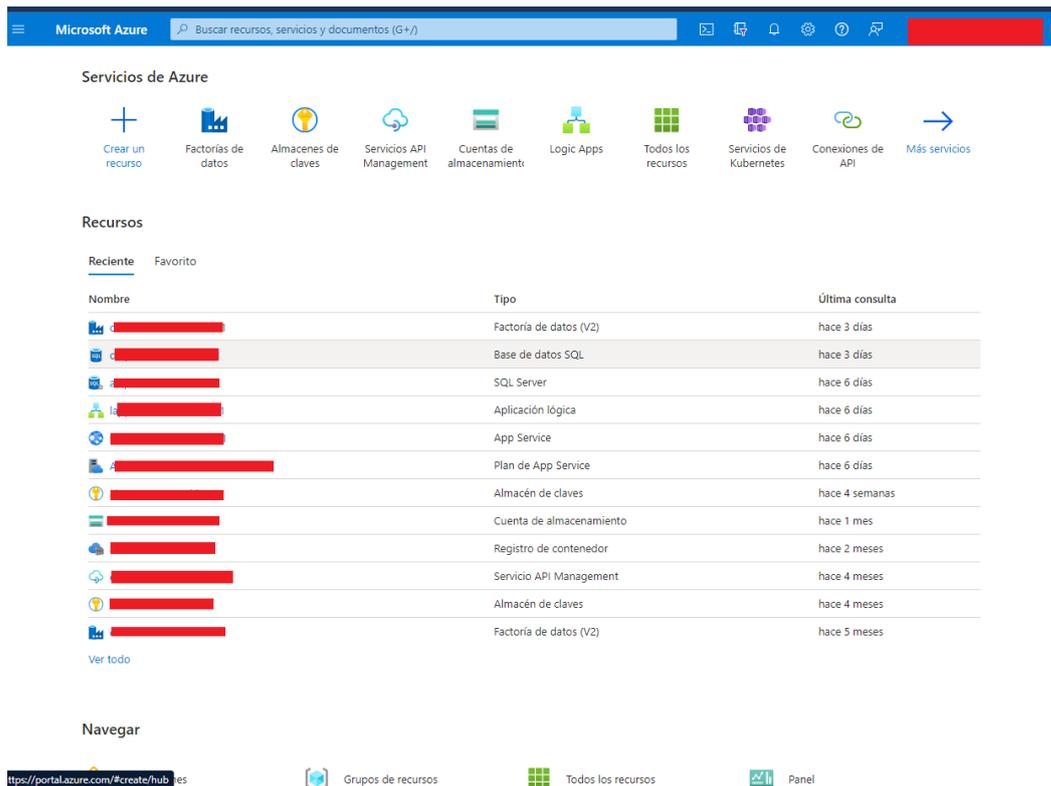


Fig. 4 Portal de Azure

5.2 Docker

Teniendo en cuenta que en este proyecto se iba a tratar con diferentes entornos, se tomó la decisión de utilizar Docker. Esta tecnología permite escoger una gran variedad de imágenes ya creadas en Dockerhub (<https://hub.docker.com>) con las características que más se ajustan a las necesidades de los distintos proyectos. Alternativamente, permite escoger estas imágenes y complementarlas con acciones adicionales en un documento Dockerfile para crear propias imágenes.

A la hora de realizar pruebas, Docker permite hacer tantas réplicas como necesitemos de manera rápida y eficiente, gracias a que el despliegue de las aplicaciones es

automático dentro de los contenedores, proporcionando una capa adicional de abstracción del sistema operativo.

5.3 Kubernetes

A medida que las aplicaciones crecen para abarcar varios contenedores implementados en varios servidores, administrarlas se hace también cada vez más complejo. Para controlar esta complejidad, Kubernetes proporciona una API de código abierto que controla la forma y el lugar donde se ejecutan esos contenedores (Fig. 5).

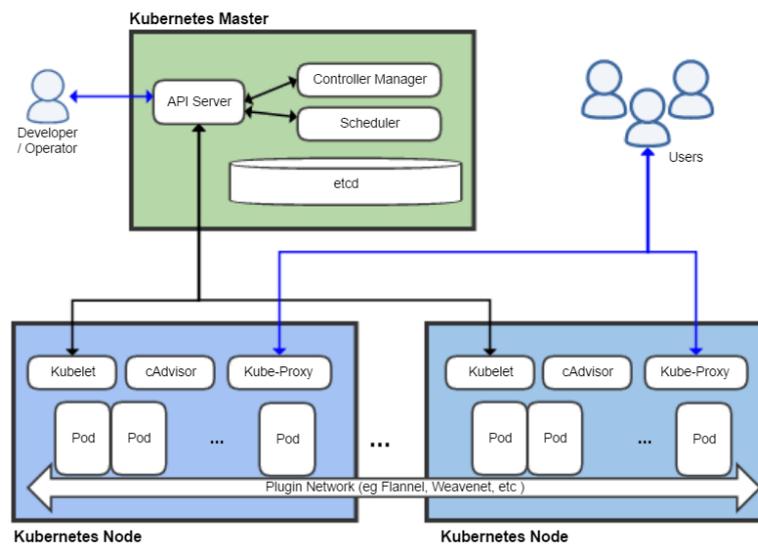


Fig. 5 Kubernetes para gestión de contenedores

6. Implementación y Configuración

Esta sección explica los diferentes aspectos técnicos de la fase de desarrollo del proyecto. Se ha seguido el orden cronológico en la que se ha ido implementando y configurando cada una de las aplicaciones que complementan el sistema.

6.1 Git

6.1.1 Preparación de Bitbucket

Para almacenar el código fuente se configuró un repositorio en la plataforma de Bitbucket. Primero, se creó un usuario con el cual los desarrolladores tendrán acceso al repositorio central mediante HTTPS o SSH. Cada usuario tendrá acceso al repositorio a través de su cuenta (Fig. 6).

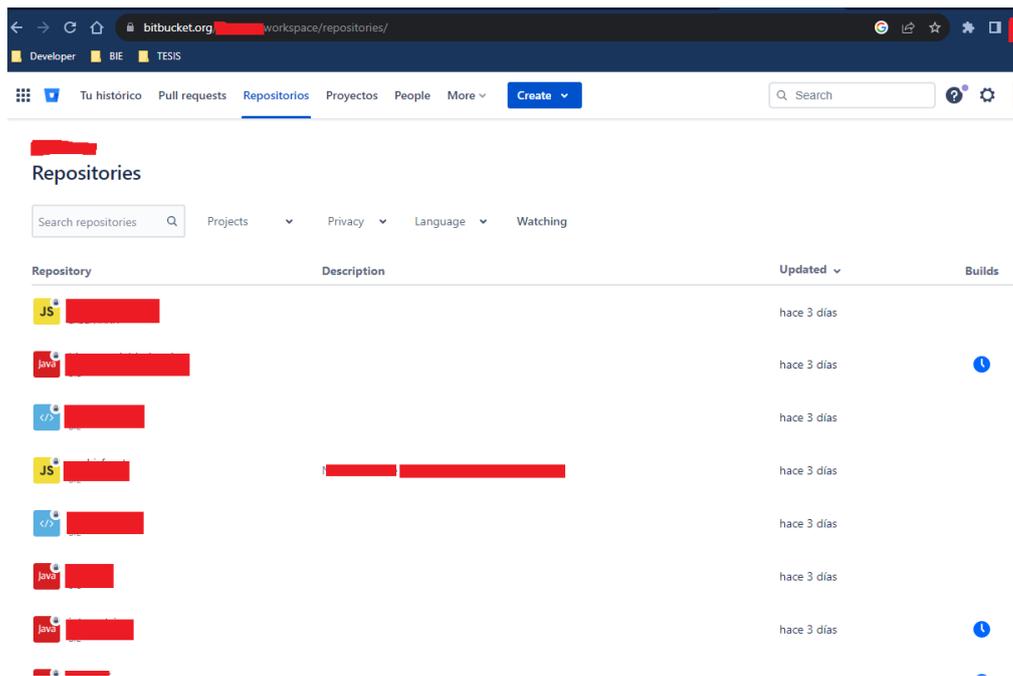


Fig. 6 Repositorios en bitbucket

6.1.2 Clonar proyecto

Una vez que los proyectos ya están en el servidor de Bitbucket, cualquier desarrollador que esté autorizado a acceder podrá clonarlo en local y empezar a colaborar en el proyecto. Desde IntelliJ se puede importar el proyecto directamente desde el menú: File > Open > Ok (Fig. 7).

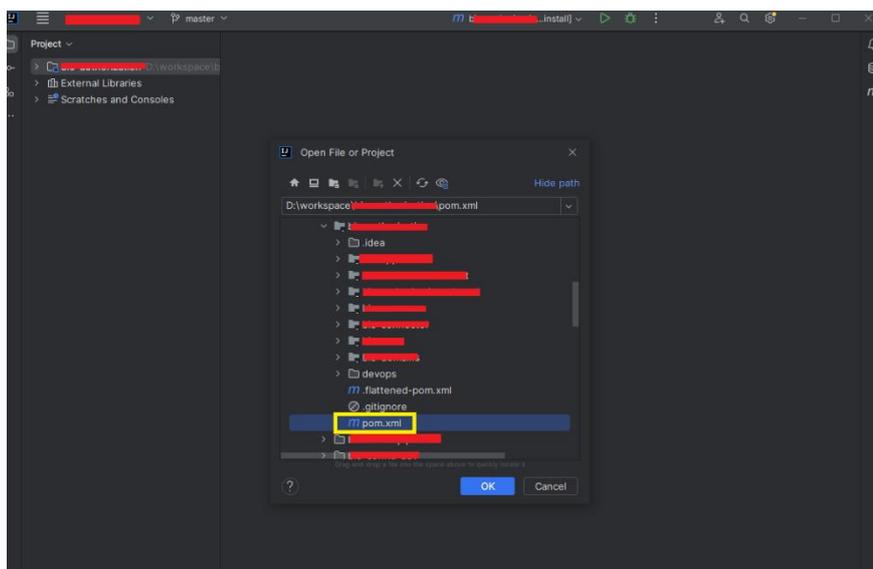


Fig. 7 Clonar proyecto localmente

6.2 Azure Pipelines

Se trata de una forma automatizada para disponibilizar nuevas funcionalidades de un software a los distintos interesados, que pueden ser usuarios finales o usuarios de prueba. Una típica dificultad dentro de todos estos pasos es la diversidad de tecnologías utilizadas para programar los sistemas, los que evolucionan particularmente rápido en el “tiempo TI”. Cada uno de estos frameworks requiere configuraciones diferentes, especialmente las páginas web, las cuales varían mucho y requieren procesos adicionales como instalación de dependencias, aplicar prefijos, minificación, pre-compilación, etc (Fig. 8).

Pipeline	Last run	Status
BACK-END [redacted] -V2	#181165 Individual CI for [redacted]	Failed (jueves <1s)
BACK-END [redacted] -V2	#181164 Individual CI for [redacted]	Failed (jueves <1s)
BACK-END [redacted]	#181163 Individual CI for [redacted]	Failed (jueves <1s)
BACK-END\BIE-NEGOTIABLE-INVOICE	#181162 Individual CI for [redacted]	Failed (jueves <1s)
BACK-END [redacted]	#180889 Individual CI for [redacted]	Failed (jueves <1s)
BACK-END [redacted]	#180888 Individual CI for [redacted]	Failed (jueves <1s)
BACK-END [redacted]	#180887 Individual CI for [redacted] release/7.8.0	Failed (jueves <1s)
[UAT] FRONT-END [redacted]	#20230727.1 • Merged in release/2023-r14--fix-v... Manually triggered for [redacted]	Success (jueves 4m 52s)
[UAT] FRONT-END [redacted]	#20230727.1 • Merged in release/2023-r14--fix-v... Manually triggered for [redacted]	Success (jueves 7m 37s)
[redacted]	#20230727.1 • Merged in feature [redacted]	Success

Fig. 8 Lista de pipelines definidos en Azure Pipelines

6.2.1 Creación del pipeline

Antes que nada, se tiene que autorizar el acceso entre la plataforma de Azure DevOps al repositorio de Bitbucket. Creamos un nuevo Pipeline en el que organizaremos nuestras tareas de manera libre: Pipelines > New pipeline > Bitbucket Cloud (Fig. 9).

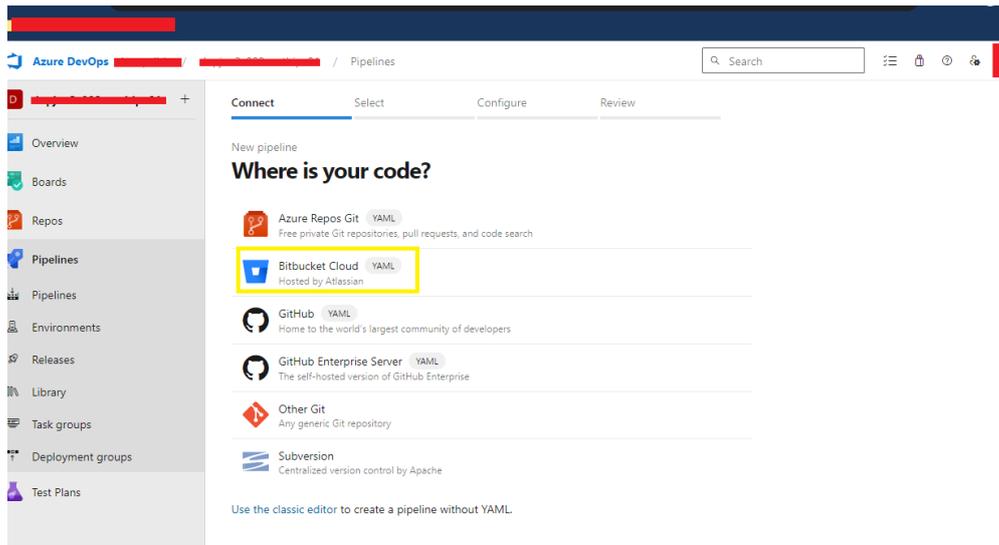


Fig. 9 Crear nuevo pipeline en Azure DevOps

6.2.2 Ejecución de pipeline

A partir del pipeline generado para los proyectos podremos, de manera automatizada, comprobar que la aplicación está compilando de manera correcta. Luego se genera la ejecución de pruebas automáticas de ser positivas, se genera el build del proyecto y se sube al Azure Container Registry. Luego se realiza el despliegue de los contenedores en el Azure Kubernetes Service según cada ambiente que se configuró para la ejecución del pipeline y de las aprobaciones realizadas (Fig. 10).

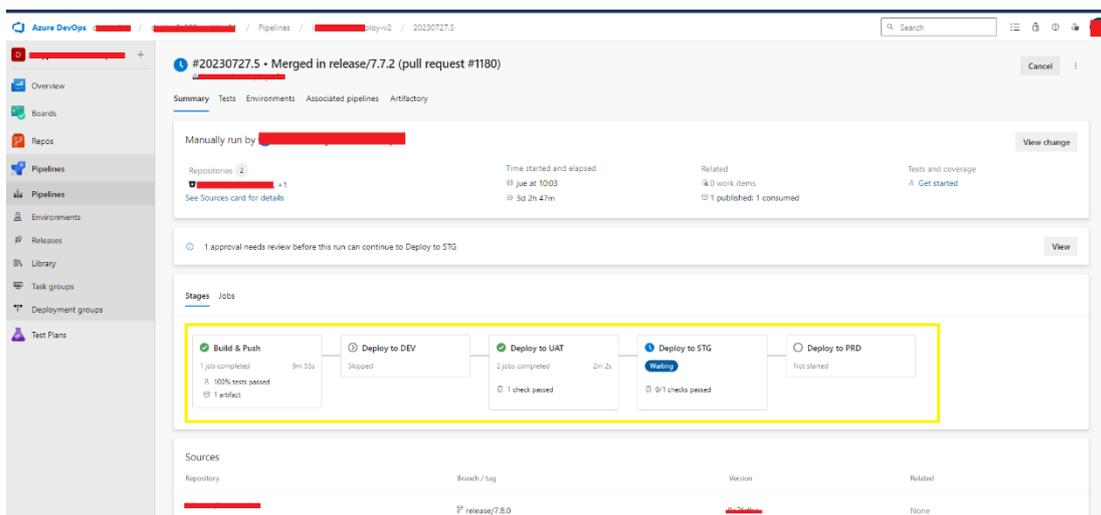
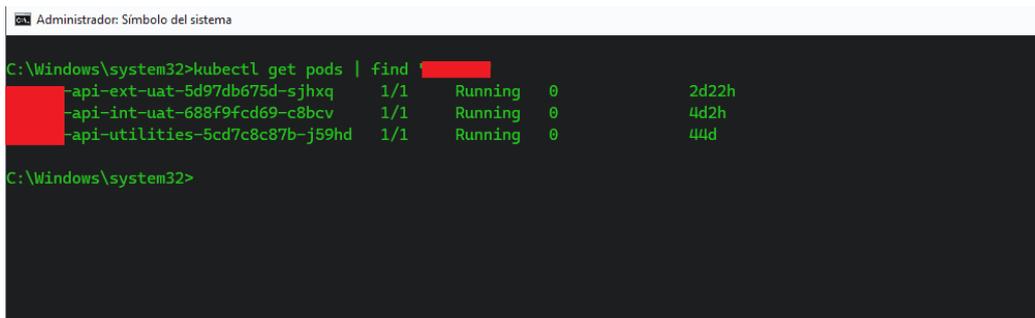


Fig. 10 Ejecutar pipeline en Azure DevOps

6.2.3 AKS

AKS (Azure Kubernetes Services) es un nuevo servicio de Azure para administrar y operar contenedores orquestados sobre Kubernetes. AKS es un servicio de contenedor gratuito que simplifica la implementación, la administración y el uso de Kubernetes como un servicio orquestador de contenedores de Kubernetes totalmente administrado.

Para los proyectos de la empresa se utilizará este servicio para el despliegue de las aplicaciones (Fig. 11).



```
Administrador: Símbolo del sistema
C:\Windows\system32>kubectl get pods | find [redacted]
[redacted]-api-ext-uat-5d97db675d-sjhxq    1/1    Running    0    2d22h
[redacted]-api-int-uat-688f9fcd69-c8bcv    1/1    Running    0    4d2h
[redacted]-api-utilities-5cd7c8c87b-j59hd    1/1    Running    0    4d
C:\Windows\system32>
```

Fig. 11 Aplicaciones desplegadas en AKS