



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

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

**DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la
Información y Comunicaciones del Ministerio de Cultura, Lima 2019**

TESIS PARA OBTENER EL GRADO ACADÉMICO DE:

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

AUTOR:

Br. Jhonny Peralta Mori (ORCID: 0000-0002-7032-5041)

ASESORA:

Dra. Liz Maribel Robladillo Bravo (ORCID: 0000-0002-8613-1882)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y Comunicaciones

Lima – Perú

2019

Dedicatoria:

A mi madre por su apoyo incondicional.

Agradecimiento:

A Dios por brindarme la oportunidad de seguir mejorando cada día más.

A mi familia porque son mi impulso a seguir para brindarles las alegrías que se merecen.

Página del jurado



DICTAMEN DE LA SUSTENTACIÓN DE TESIS

EL / LA BACHILLER (ES): **PERALTA MORI, JHONNY**

Para obtener el Grado Académico de *Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información*, ha sustentado la tesis titulada:

DEVOPS EN LA ENTREGA CONTINUA DE LA OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES DEL MINISTERIO DE CULTURA, LIMA 2019

Fecha: 16 de agosto de 2019

Hora: 5:00 p.m.

JURADOS:

PRESIDENTE: Dr. César Humberto Del Castillo Talledo

Firma: 

SECRETARIO: Dra. Paula Viviana Liza Dubois

Firma: 

VOCAL: Dra. Liz Maribel Robladillo Bravo

Firma: 

El Jurado evaluador emitió el dictamen de:

..... *Aprobado por mayoría*

Habiendo encontrado las siguientes observaciones en la defensa de la tesis:

.....
.....
.....

Recomendaciones sobre el documento de la tesis:

- *Mejoras APA*
- *Revisar conclusiones y recomendaciones*
- *mejorar resultados*

Nota: El tesista tiene un plazo máximo de seis meses, contabilizados desde el día siguiente a la sustentación, para presentar la tesis habiendo incorporado las recomendaciones formuladas por el jurado evaluador.

Declaratoria de autenticidad

Yo, Jhonny Peralta Mori, estudiante de la Escuela de Posgrado, del programa Maestría en Ingeniería de Sistemas con Mención en Tecnologías de la Información, de la Universidad César Vallejo, Sede Lima Norte; presento mi trabajo académico titulado: DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, Lima 2019, en folios para la obtención del grado académico de Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información.

Por tanto, declaro lo siguiente:

- He mencionado todas las fuentes empleadas en el presente trabajo de investigación, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes, de acuerdo con lo establecido por las normas de elaboración de trabajos académicos.
- No he utilizado ninguna otra fuente distinta de aquellas expresamente señaladas en este trabajo.
- Este trabajo de investigación no ha sido previamente presentado completa ni parcialmente para la obtención de otro grado académico o título profesional.
- Soy consciente de que mi trabajo puede ser revisado electrónicamente en búsqueda de plagios.
- De encontrar uso de material intelectual ajeno sin el debido reconocimiento de su fuente o autor, me someto a las sanciones que determinen el procedimiento disciplinario.

Lima, 16 de agosto de 2019



Jhonny Peralta Mori

70143675

Índice

Dedicatoria:	ii
Agradecimiento:	iii
Página del jurado	iv
Declaratoria de autenticidad	v
Índice	vi
Índice de tablas	viii
Índice de figuras	ix
Resumen	x
Abstract	xi
I. Introducción	1
II. Método	14
2.1 Tipo y diseño de investigación	14
2.2 Operacionalización de variables	15
2.3 Población, muestra y muestreo	18
2.3.1 Población	18
2.3.2 Muestra	18
2.4 Técnicas e instrumentos de recolección de datos, validez y confiabilidad	19
2.5 Procedimiento Método de análisis de datos	20
2.6 Método de análisis de datos	20
2.7 Aspectos éticos	20
III. Resultados	21
IV. Discusión	29
V. Conclusiones	35
VI. Recomendaciones	36

VII. Propuesta	37
Referencias	38
Anexos	44
Matriz de consistencia	45
Instrumentos	47
Validez de los instrumentos	53
Matriz de datos (Excel y/o spss)	56
Proyecto / intervención/ Aplicación	62
Autorización	71
Procesos	72

Índice de tablas

Tabla 1: Identificación de variables	16
Tabla 2: Operacionalización	17
Tabla 3: Estadísticos descriptivos de la Frecuencia de despliegue	21
Tabla 4: Estadísticos descriptivos de los Cambios Fallidos	22
Tabla 5: Estadísticos descriptivos de los Cambios sin Errores	23
Tabla 6: Prueba de normalidad de la Frecuencia de despliegue	24
Tabla 7: Prueba de normalidad de los Cambios Fallidos	25
Tabla 8: Prueba de normalidad de los Cambios sin Errores	25
Tabla 9: Estadístico de Prueba T de Frecuencia de Despliegue	26
Tabla 10: Estadístico de Prueba T de Frecuencia de Despliegue	27
Tabla 11: Estadístico de Prueba T de Frecuencia de Despliegue	27
Tabla 12: Hipotesis General	28

Índice de figuras

Figura 1: Media de Frecuencia de Despliegue	22
Figura 2: Media de Cambios Fallidos	23
Figura 3: Media de Cambios sin Errores	24
Figura 4: Maquina Virtual y Contenedor	30
Figura 5: Docker que es un contenedor	31
Figura 6: GitLab que es un sistema de versión de contenidos	32
Figura 7: Composer que es un administrador de paquetes	33
Figura 8: Drush administra por consola a Drupal	33
Figura 9: Drupa que es un Framework toda vez que es modular, multipropósito y muy configurable	34

Resumen

La presente investigación tuvo como objetivo demostrar que DevOps influye positivamente en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura. El estudio estuvo orientado a los problemas que aquejaban en los despliegues pasados a producción de las aplicaciones web, en donde se pudo demostrar que DevOps aumenta la frecuencia de despliegue, disminuye en los cambios fallidos y aumenta los cambios sin errores.

El tipo de estudio fue aplicada del diseño experimental de tipología pre experimental, para realizar el contraste de los objetivos y determinar si es aceptada o rechazada se analizó el antes y después de las variables luego de haber sido expuestas al estímulo, se realizó las pruebas de hipótesis mediante la prueba paramétrica de T de student.

Con DevOps se logró aumentar la frecuencia de despliegue en un 49.26%, disminuir los cambios fallidos en un 57.42% y aumentar los cambios sin errores en un 25.6%. en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura

Palabras clave: DevOps, Entrega Continua, Despliegue

Abstract

This research aimed to demonstrate that DevOps positively influences the Continuous Delivery of the General Office of Statistics and Information Technology and Communications of the Ministry of Culture. The study was oriented to the problems that faced the deployment of production of web applications, where it was shown that DevOps increases the frequency of deployment, decreases in failed changes and increases changes without errors.

The type of study was applied from the experimental design of pre-experimental typology, to perform the contrast of the objectives and determine if it is accepted or rejected, the before and after the variables were analyzed after being exposed to the stimulus, the tests of Hypothesis using the parametric test of student's T.

With DevOps it was possible to increase the frequency of deployment by 49.26%, decrease the failed changes by 57.42% and increase the changes without errors by 25.6%. in the Continuous Delivery of the General Office of Statistics and Information Technology and Communications of the Ministry of Culture

Keywords: DevOps, Continuous Delivery, Deployment

I. Introducción

Hoy en día hay una ola entorno a DevOps, al utilizarlo se vuelven simples los procesos para que se sean cada vez más programables y dinámicos, ello influye en una mayor frecuencia de pases a producción, acorta el tiempo de despliegues, menora los errores de lanzamientos y en caso surjan menora el tiempo de recuperación. Las empresas como IBM, Amazon AWS, Google Compute Cloud, Microsoft Azure, Google Apps Engineer, Jelastic entre otras como Flickr que desarrolló un enfoque DevOps para apoyar un requisito de negocio de diez despliegues por día.

Es así como tenemos 3 ejemplos Relevantes: (1) Amazon: Se estima que en el 2011 operaba a más de 137.000 usuarios por semana, moviendo beneficios anuales de una suma de 34.000 millones de dólares, lo increíble es que cada 11,6 segundos en promedio realiza despliegues versionados en producción, con el impacto de afectar a 30.000 servidores. (2) Flickr: Aloja de los usuarios más de 3.000 millones de fotos, lo cual lo logra porque cada minuto cuenta en promedio 3.000 fotos subidas, con el gran movimiento de fotos puede realizar varios despliegues diarios. (3) Facebook: Cuenta con más de 650 millones de usuarios activos en aumento, no es de sorprender que en el planeta por 13 personas 1 tiene facebook, al realizar cambios en el código al ser compilado se crea un archivo binario con el tamaño de 1,5Gb, el cual si es cambios menores se despliega en producción al menos con una frecuencia diaria, si son cambios grandes se realiza una vez a la semana. Entonces hablamos de empresas con lanzamientos muy frecuentes pueden requerir de sensibilización y orientación DevOps.

En la presente investigación, donde la Realidad problemática, en el Ministerio de Cultura en su constante innovación para brindar mejores servicios a los ciudadanos se encuentra con problemas que aquejan entre entregar un producto final o en el momento de desplegar los proyectos de sistemas web, donde pueden surgir errores que retrasan el lanzamiento o pase a producción, hay eventos que suceden repetitivos impidiendo algunas funcionalidades del sistema, sucesos que requieren leer todo un manual de usuario, que se podrían optimizar para reducir el impacto hacia el ciudadano. En el Ministerio de Cultura, en la Oficina General de Estadística y Tecnología de la Información y Comunicaciones, en la Oficina de Desarrollo Tecnológico se plantea DevOps para resolver problemas y poder

desplegar rápidamente sistemas, aplicaciones app, consultas para mejorar el servicio a los ciudadanos.

En Antecedentes Internacionales: (a) En el año 2018, Luis Alberto Iñiguez Sánchez en tesis de la maestría denominada “Arquitectura para la entrega continua de software con despliegue en contenedores”, en esta investigación, al carecer de una única arquitectura para que el software sea desarrollado, se ha visto generado que los despliegues de las versiones nuevas en las aplicaciones, son una tarea grande además de complicada, teniendo con facilidad errores en el software que se pone en el entorno de producción, con ello dificultando que la calidad sea asegurada. Por ende, a la necesidad se ha planteado la implementación de un pipeline de entrega continua de software, de esta manera el equipo de desarrollo será potenciado para mitigar los riesgos a cada despliegue de producto.

(b) En el año 2015, López Aguilar, Francisco Josué en tesis de maestría denominada “Aprovisionamiento automático de infraestructura y de configuración de la infraestructura. El caso Adblock Plus”, en esta investigación, presenta una solución que involucra software de código abierto que permite utilizar las mejores prácticas del desarrollo de software aplicadas al mantenimiento, configuración y administración de la infraestructura.

(c) En el año 2017, Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. en DevOps. IEEE Software, menciona sobre la base de prácticas ágiles y ágiles, DevOps significa automatización de extremo a extremo en el desarrollo y la entrega de software. Casi nadie podrá abordarlo con un enfoque de estilo de libro de cocina, pero la mayoría de los desarrolladores se beneficiarán al conectar mejor los silos aislados de desarrollo y operaciones. Existen muchas herramientas DevOps que pueden ayudarles a hacer esto.

(d) En el año 2016, Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. en What is DevOps? In Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops menciona que DevOps, la combinación de Desarrollo y Operaciones, es una nueva forma de pensar en el dominio de la ingeniería de software que recientemente recibió mucha atención. Dado que DevOps es un término nuevo y un concepto novedoso recientemente introducido, aún no se ha logrado un entendimiento común de lo que implica.

En consecuencia, las definiciones de DevOps of-ten solo representan una parte que es relevante para el concepto. Objetivo: Este estudio pretende caracterizar DevOps mediante la exploración de los componentes centrales de las definiciones de DevOps reportadas en la literatura, especificando prácticas explícitamente propuestas para DevOps e investigando las similitudes y diferencias entre DevOps y otros métodos existentes en ingeniería de software.

Método: Se realizó un estudio de mapeo sistemático que utilizó seis bases de datos electrónicas: IEEE, ACM, Inspec, Sco-pus, Wiley Online Library y Web of Science. Resultado: se seleccionaron 44 estudios que informan una definición de DevOps, 15 estudios que expresan explícitamente las prácticas de DevOps y 15 estudios que indican cómo DevOps está relacionado con otros métodos existentes. En algunos casos, los documentos establecieron una combinación de una definición, prácticas y relaciones con otros métodos, el número total de estudios primarios fue de 49.

Conclusión: Propusimos una definición para DevOps que puede superar las inconsistencias en las diversas definiciones existentes. niciones de estudios de investigación individuales. Además, se han presentado las prácticas explícitamente propuestas para DevOps, así como la relación con otros métodos de desarrollo de software.

(e) En el año 2016, Ravichandran, A., Taylor, K., & Waterhouse, P. en DevOps for Digital Leaders. DevOps for Digital Leaders menciona que este libro proporciona a los líderes digitales responsables del desarrollo rápido de alta calidad de aplicaciones de software en una guía concisa para diseñar, implementar, medir y mejorar los programas DevOps que se adaptan a sus organizaciones. En DevOps para líderes digitales, una experiencia colectiva profunda en ambos. Los lados de la división de devops informan sobre el liderazgo de pensamiento global y los conocimientos penetrantes de los autores, los tres de los cuales son líderes de DevOps de cartera cruzada en CA Technologies. Aruna Ravichandran, Kieran Taylor y Peter Waterhouse analizan los beneficios organizativos, los costos, las libertades y las limitaciones de DevOps. Trazan la estrategia coordinada de cambio organizativo, métricas, pensamiento ágil e inversión que una empresa debe emprender para aprovechar todo el potencial de DevOps y alcanzar el punto óptimo donde las implementaciones de código aceleradas aumentan la satisfacción del cliente, los ingresos y la rentabilidad.

Los líderes digitales tienen la obligación de salvar la desconexión entre desarrolladores y operadores si sus organizaciones quieren sobrevivir y prosperar en un mundo de negocios cada vez más diferenciado por el grado en que el desarrollo dinámico de software de aplicaciones armoniza con la resistencia y confiabilidad operativas. Este breve libro aplica la perspectiva de DevOps al desafío competitivo, al que se enfrenta cada organización de TI de alto rendimiento en la actualidad, de integrar y automatizar las herramientas, los procesos y las técnicas de código abierto, la nube y la empresa en la totalidad del ciclo de vida del desarrollo de software, desde los requisitos hasta el lanzamiento.

(f) En el año 2016, Zhu, L., Bass, L., & Champlin-Scharff, G. en *DevOps and Its Practices*. IEEE Software menciona que DevOps tiene como objetivo disminuir el tiempo entre realizar un cambio en el sistema y colocar el cambio en producción normal, al tiempo que garantiza una alta calidad. Los temas del artículo en este tema temático incluyen el uso de DevOps para migrar a microservicios, la adopción de DevOps y las herramientas de DevOps.

(g) En el año 2015, Lwakatare, L. E., Kuvaja, P., & Oivo, M. en *Dimensions of devOps*. In *Lecture Notes in Business Information Processing* menciona que DevOps ha sido identificado como un aspecto importante en el paradigma de despliegue continuo en comunidades de profesionales y círculos de investigación académica. Sin embargo, poco se ha presentado para describir y formalizar lo que constituye. La falta de tal comprensión significa que el fenómeno no se comunicará de manera efectiva y su impacto no se entenderá en esas dos comunidades. Este estudio investiga los elementos que caracterizan el fenómeno de DevOps mediante una encuesta bibliográfica y entrevistas con practicantes que participan activamente en el movimiento DevOps. Se identifican cuatro dimensiones principales de DevOps: colaboración, automatización, medición y monitoreo. Se desarrolla un marco conceptual inicial para comunicar el fenómeno a los profesionales y a la comunidad científica, así como para facilitar información para futuras investigaciones.

(h) En el año 2017, Nielsen, P. A., Winkler, T. J., & Nørbjerg, J. en *Closing the IT development-operations gap: The devops knowledge sharing framework*. In *CEUR Workshop Proceedings* menciona que aunque DevOps ha sido anunciado como un

paradigma novedoso para superar los límites tradicionales entre los equipos o grupos de Desarrollo de TI (Dev) y Operaciones de TI (Ops), muchas organizaciones de TI carecen de orientación sobre cómo implementar este paradigma en la práctica. Esta investigación de diseño proporciona un marco que puede ayudar a las organizaciones a evaluar no solo su estado en el cumplimiento de los principios, prácticas y uso de herramientas DevOps recomendados.

Dado el papel central del intercambio de conocimientos en la entrega de software, nuestro marco también incluye la conversión de conocimientos requerida entre los equipos de desarrollo y operaciones. basándose en las dimensiones del modelo SECI ampliamente reconocido. Evaluamos el marco de intercambio de conocimientos de DevOps (DOKS) propuesto en el contexto de una pequeña empresa de servicios de TI y una gran empresa de servicios financieros. Nuestros hallazgos establecen que el marco DOKS puede ayudar a las organizaciones mediante la sensibilización de los dos equipos en cuanto a los elementos cruciales de DevOps. Además, discutimos cómo los enfoques de implementación de DevOps pueden diferir entre las organizaciones de TI más pequeñas y más grandes.

(i) En el año 2017, Shahin, M., Zahedi, M., Babar, M. A., & Zhu, L. en *Adopting Continuous Delivery and Deployment* menciona que las prácticas de entrega e implementación continuas (CD) apuntan a entregar las características del software de manera más frecuente y confiable. Si bien se han realizado algunos esfuerzos para estudiar diferentes aspectos de las prácticas de los CD, se ha informado un poco de trabajo empírico sobre el impacto de los CD en las estructuras del equipo, la colaboración y las responsabilidades de los miembros del equipo. Objetivo: Nuestro objetivo es investigar empíricamente cómo se organizan los equipos de Desarrollo (Dev) y Operaciones (Ops) en la industria del software para adoptar prácticas de CD. Además, exploramos el impacto potencial de practicar CD en la colaboración y las responsabilidades de los miembros del equipo. Método: Realizamos un estudio empírico de métodos mixtos, que recopiló datos de 21 entrevistas semiestructuradas en profundidad en 19 organizaciones y una encuesta con 93 profesionales del software.

Resultados: Hay cuatro tipos comunes de estructuras de equipo (es decir, (1) equipos de desarrollo y operaciones separados con mayor colaboración; (2) equipos de desarrollo y operaciones separados con facilitadores en el medio; (3) equipo de operaciones pequeño con

más responsabilidades para el equipo de desarrollo (4) (no hay un equipo de operaciones visible) para organizar los equipos de desarrollo y operaciones para iniciar y adoptar prácticas de CD de manera efectiva. Nuestro estudio también proporciona información sobre cómo las organizaciones de software realmente mejoran la colaboración entre los equipos y los miembros del equipo para practicar CD. Además, destacamos nuevas responsabilidades y habilidades (por ejemplo, habilidades de monitoreo y registro), que son necesarias a este respecto.

(j) En el año 2014, Michelsen, J. en *A Pragmatic Guide to Getting Started with DevOps* menciona que DevOps no es un producto, ni siquiera una tecnología en particular. DevOps es una metodología, en donde las separadas desarrollo de software (Dev) y producción y operaciones (Ops) se unen en un proceso único, integrado y continuo. DevOps se trata de romper las barreras entre Dev y Ops. Aprovecha a las personas, los procesos y la tecnología para estimular la colaboración y la innovación en todo el proceso de desarrollo y lanzamiento de software. Dev y Ops deben actuar y sentir que son un solo equipo. Este documento actúa como una guía sobre DevOps, si es adecuado para usted, los 7 signos de un proceso disfuncional y donde usted también puede iniciar la transformación de sus devops.

(k) En el año 2013, Onno Dijkstra en la Master Thesis denominada “La extensión de la Disciplina de Desarrollo Ágil de Despliegue, La necesidad de un enfoque holístico” desarrollada en la Utrecht University, Utrecht - Países Bajos, en esta investigación identifica los diferentes caminos que se pueden elegir para poner en práctica DevOps para las empresas que utilizan Scrum como su método de desarrollo actual y experimentan los mismos problemas. DevOps propone soluciones a los conductores en el interés de la empresa. La investigación muestra que la razón es de utilizar DevOps proporcionando una explicación en profundidad lo que los conductores provocan la empresa para implementar esta extensión. Además, los caminos implementaciones se proporcionan para ayudar a otras empresas que se enfrentan a los mismos problemas con su proceso de desarrollo. La declaración formal problema para este proyecto de investigación se formula de la siguiente manera: ¿Cómo puede la organización de TI con el apoyo en la implementación de DevOps?

(1) En el año 2012, Gaute Borgenholt en la Master Thesis denominada “Audición: un marco de control y pruebas de calidad orientado a DevOps para entornos de nube” desarrollada en la University of Oslo, Oslo - Noruega, en esta investigación menciona que al centralizar el software, las interrupciones en el servicio afectan a todos los usuarios. Además, cualquier cambio en el software es extremadamente lento, con el cepillado, las pruebas en múltiples entornos, aplicación de prueba, a continuación, la aplicación de producción. Los usuarios esperan con frecuencia libera de nuevas características usuarios esperan nuevas características publicadas con regularidad y frecuencia, sin degradación de la calidad del servicio. Esta solicitud para la entrega continua de nuevas características, han llevado a la creación de DevOps. Un paradigma de desarrollo con un enfoque en la automatización y las entregas continuas. Con todo el sistema de la administración diaria tareas y todos los pasos en el mantenimiento de un servicio, el administrador del sistema, además de tener que liberar nuevas características con intervalos más cortos y más cortos para atraer nuevos usuarios y mantener la corriente.

Para sustentar la presente investigación utilizamos los siguientes teóricos. En donde la variable (I) DevOps, "DevOps, un acrónimo de desarrollo y operaciones, medios para cerrar las brechas [entre el desarrollo y las operaciones] por alinear incentivos y compartir enfoques de procesos y herramientas" - DevOps para desarrolladores, Michael Hüttermann (2012). Por encima de todo, la aplicación del enfoque DevOps es un cambio de mentalidad. Es esencial para construir un equipo de Devs y Operaciones, que está alineado con incentivos compartidos. - Hüttermann (2012).

DevOps es una respuesta puntual, a la conciencia creciente para realizar la desconexión entre lo que se considera tradicionalmente todas las actividades de desarrollo y todas las actividades de operaciones. Con la menuda desconexión conlleva a la manifestación de conflicto y la ineficiencia". - Damon Edwards, "¿Qué es DevOps?". El enfoque DevOps es una capacidad empresarial esencial para la entrega continua de software, al permitir que las organizaciones aprovechen las oportunidades del mercado y tarden menos en obtener la retroalimentación de los clientes.- IBM; “ DevOps: el enfoque de IBM”.

Payal Chakravarty. (2014). The DevOps Scorecard. mayo 5, 2019, de Devops Sitio web: <https://devops.com/devops-scorecard/>, menciona que en el cuadro de mando DevOps debe contener estos 9 métricas para rastrear el éxito del equipo DevOps: (1) Frecuencia de despliegue: ¿Con qué frecuencia se nos despliega código y obtener nuevo código en manos de nuestros clientes? Esta métrica debería tender hacia arriba o permanecer estables de semana en semana. Ejemplo: Dos veces a la semana, 50 veces al día. (2) Cambiar el volumen: Para cada despliegue cuántas historias de usuario y nuevas líneas de código se nos envió? Ejemplo: 3 nuevas características por día, promedio de 500 líneas de código nuevo por semana. Otro parámetro a considerar además de volumen es la complejidad del cambio. (3) Plazo de ejecución (de Dev implementar): ¿Cuánto tiempo se tarda en promedio para obtener el código de desarrollo integral a través de un ciclo de pruebas A / B a 100% de despliegue y actualización en la producción? El plazo de ejecución se debe reducir en el equipo consigue una mejor sujeción del ciclo de vida.

(4) Porcentaje de implementaciones fallidas: ¿Qué porcentaje de las implementaciones no causando un apagón o una reacción negativa al usuario? Esta métrica debería disminuir con el tiempo. Ejemplo: 9 despliegues% fracasó este mes frente a un 15% el mes pasado. Esta métrica debería revisarse en combinación con el cambio de volumen. Si el cambio de volumen es bajo o sigue siendo el mismo, pero el porcentaje de implementaciones fallidas aumentado, entonces tal vez una disfunción en alguna parte. (5) La media de tiempo de recuperación: Cuando hicimos fracasamos, ¿cuánto tiempo nos tarda en recuperar? Esto es un verdadero indicador de lo bien que estamos consiguiendo con la gestión del cambio y esto, lo ideal sería reducir el tiempo. Usted puede contar con algunos picos en este número debido a problemas complejos no encontrado antes. Ejemplo: En un promedio que tomó el equipo de 15 minutos para resolver cada uno la semana pasada a 14 minutos esta semana.

(6) Cliente volumen de entradas: El número de alertas generadas por los clientes indicar problemas en el servicio. Este es un indicador básico de la satisfacción del cliente. Ejemplo: 54 entradas se generaron esta semana en lugar de 38, mientras que el volumen de usuarios se mantuvo estable no es una buena cosa. (7) Porcentaje de cambio en el volumen de Usuario: Número de nuevos usuarios que se registran, interactuando con mi servicio y la

generación de tráfico. A medida que nuevos usuarios se registran es mi infraestructura capaz de manejar la demanda? Ejemplo: Esta semana, el número de clientes con púas en un 30% debido a un evento externo que causa volumen de solicitudes para subir.

(8) Disponibilidad: ¿Cuál es el tiempo de actividad global de mi servicio y me violo ningún SLA? Ejemplo: 99.9% consistentemente durante los últimos 3 meses, incluso con el cambio en el volumen de usuarios. (9) Rendimiento (Tiempo de respuesta): ¿Mi servicio funcionando dentro de mis umbrales predeterminados? Esta métrica debería permanecer estable con independencia de cambio% en el volumen de usuario o de cualquier nueva implementación. Ejemplo: Sub 5 segundos el tiempo de respuesta de todas las geografías y dispositivos.

Según Chris Riley. (2015). Metrics for DevOps. mayo 5, 2019, de Devops Sitio web: <https://devops.com/metrics-devops/>, menciona que la mayoría de las métricas DevOps se dividen en 3 categorías generales: (1) La gente: La gente es una parte intrínseca de cualquier proceso DevOps. Métricas orientadas a las personas miden cosas tales como el volumen de negocios, capacidad y tiempo de respuesta. Comience siempre con la gente. Son el elemento más difícil de cualquier elemento, y su influencia es a veces difícil de detectar. (2) Proceso: De alguna manera, DevOps es todo acerca de proceso - ciclo el despliegue continuo / operaciones / soporte es un conjunto continuo de procesos entrelazados. Pero algunas métricas son más claramente orientado al proceso que otros, en particular los que implican la entrega continua, respuesta y reparación. Tiempo de plomo-Development necesario para el despliegue, por ejemplo, es una respuesta en frecuencia y tiempo de despliegue en gran medida orientado al proceso métrica, como lo son. Métricas de proceso puede ser una medida de la velocidad (¿Dónde están los cuellos de botella, y es el proceso en sí mismo un cuello de botella?), Adecuación (¿Son todas las medidas pertinentes?), Efectividad (¿Consigue el trabajo hecho?), O eficiencia (¿Son los pasos en la secuencia óptima? ¿hay un flujo suave en el proceso?).

(3) Tecnología: Métricas de tecnología también desempeñan un papel importante en DevOps, medir cosas tales como el tiempo de actividad (¿Qué porcentaje de las veces es el sistema que ejecuta? ¿Qué pasa con la red, y aplicaciones de apoyo?) y tasa de fracaso (¿Cuál es el porcentaje de implementaciones fallidas, cambios o unidades?).

Por supuesto, muchas DevOps métricas implican las tres categorías, en mayor o menor grado. Tal vez la forma más fácil de ver cómo las 4 métricas desempeñan en la práctica es buscar en las métricas clave utilizada por Puppet Labs (Puppet no es un cliente mío, ni soy un cliente de ellos): (1) Despliegue (o cambiar) Frecuencia: DevOps prácticas hacen despliegue frecuente o continuo posible; grandes, los sitios web de alto tráfico y los servicios basados en la nube hacen que sea una necesidad. Con respuesta rápida y el desarrollo de lotes pequeños, software actualizado se puede implementar cada pocos días o incluso varias veces al día. En un entorno DevOps, frecuencia de despliegue puede ser una medida directa o indirecta de tiempo de respuesta, la cohesión del equipo, las capacidades de desarrollo, eficacia herramienta de desarrollo, y DevOps generales eficiencia del equipo.

(2) Cambie Plazo de ejecución: El tiempo transcurrido desde el inicio de un ciclo de desarrollo (el primer código nuevo) para el despliegue es el momento de cambio de líder. Es una medida de la eficiencia del proceso de desarrollo, de la complejidad del código y de los sistemas de desarrollo, y también (como frecuencia de despliegue) de las capacidades del equipo y de desarrollador. Si el tiempo de cambio de plomo es demasiado largo, puede ser una indicación de que el proceso / despliegue desarrollo es ineficaz en ciertas etapas, o que incluye los cuellos de botella de rendimiento.

(3) Cambio Porcentaje de averías: Uno de los principales objetivos de DevOps es convertir despliegues rápidos y frecuentes en un asunto cotidiano. Huelga decir que, para tales despliegues a tienen un valor, la tasa de fracaso debe ser bajo. Se debe, de hecho, disminuir con el tiempo, ya que la experiencia y las capacidades de los equipos DevOps aumentan. Una tasa de fracaso cada vez mayor, o uno que es alto y no baja con el tiempo, es una buena indicación de problemas en el proceso general DevOps.

(4) Tiempo medio para recuperar (MTTR): Este es el momento de la falla en la recuperación de ese fracaso. Por lo general es una buena medida de la capacidad del equipo, y al igual que la tasa de fracaso, se debe mostrar una disminución general en el tiempo (lo que permite tiempos de recuperación más largos ocasionales cuando el equipo se encuentra con un problema técnico no familiar). MTTR también puede verse afectada por cosas tales como código (o plataforma) la complejidad, el número de nuevas características está

aplicando, y cambios en el entorno operativo (como la migración a un nuevo servidor de la nube). Sobre la base de estos indicadores, el equipo DevOps ideales produciría despliegues rápidos y frecuentes mínima (y decreciente) tasa de fracaso y una corta (y encogimiento) el tiempo de recuperación. En la práctica, por supuesto, puede haber factores que van en contra de estas tendencias - menos necesidad u oportunidad para los despliegues frecuentes, por ejemplo, o los frecuentes cambios en las condiciones de funcionamiento o requisitos.

En donde la variable (II) La Entrega Continua, usa en conjunto los patrones de buenas prácticas en el desarrollo de software, en donde busca la manera de construcción enfocado para ser liberado de una forma ágil, segura y en cualquier momento en entornos productivos, volviéndose una disciplina, de esta manera el objetivo es conseguir una puesta en producción con las siguientes características: rápida, frecuente, reproducible y sobre todo asumiendo pocos riesgos. Eder Castro Lucas (2014).

En donde la entrega continua cuenta con la dimensión eficacia y los 3 indicadores: (1) Frecuencia de despliegue: en donde la total flexibilidad, se adopta nuevos cambios con entregas más frecuentes, con ello se controla y maneja de mejor manera los entregables. (2) Cambios Fallidos: Cada cambio desde la primera línea de código, pasa por el mismo conjunto de pasos que nos garantizan la robustez de la entrega, con ello la entrega se convierte en un proceso transparente y predecible, de esta manera su frecuente utilización le otorga estabilidad y confiabilidad, finalmente evitando las largas e interminables puestas en producción; y, por lo tanto, evitando las sorpresas de última hora. (3) Cambios sin errores: En las fases tempranas los problemas que aparecen en un mismo proceso se abordan a tiempo para resolver, obteniéndose una mayor trazabilidad, por ende al detectar antes se solucionan antes, consiguiendo en producción menos errores. De donde se puede deducir que frecuencia de despliegue es el numero de despliegues solicitados entre los realizados, así mismo los cambios fallidos son el número de cambios fallidos entre el total de cambios cerrados, finalmente los cambios sin errores son el numero de cambios entre total de cambios cerrados.

Con ello los cambios muy grandes y costosos son de Mayor incertidumbre en la entrega, a comparación de los cambios pequeños y controlados son de Menor incertidumbre en la entrega. Liberar las versiones de manera continuamente, permite la posibilidad de reaccionar rápidamente a los cambios de mercado como a las necesidades de clientes, así mismo de su

organización; de una manera ágil, y sin la necesidad de tener que invertir en diferentes departamentos con una enorme cantidad de tiempo. Puesto que es menor el tiempo en desplegar y entregar en producción, toda vez el proceso que tenemos es automatizado. Así mismo nos permite la obtención de feedback rápidamente de como estos cambios afectan al negocio por parte de los usuarios.

De lo expuesto, se plantean la problemática siguiente: Problema principal: PA: ¿Cómo DevOps influye en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura?. Problemas secundarios: P1: ¿En qué medida el uso de DevOps influye en el Cambios Fallidos de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura?. P2: ¿En qué medida el uso de DevOps influye en la Frecuencia de Despliegue de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura?.

Así mismo la Justificación de la presente investigación: Justificación Tecnológica: La situación actual de los requisitos nuevos de los usuarios y exigencias en pases a producción, con los dramáticos cambios tecnológicos en el tratamiento de información, son razones importantes, entre otras que sugieren la realización de nuevas formas de entrega continua. Justificación Económica: Al tener grandes pases a producción se encuentra con mayores riesgos que la fallar demanda tiempo y con ello sobre costos, al aplicar DevOps se minimizan los errores ahorrando significativamente costos. Justificación Institucional: Siendo de gran importancia brindar mejores servicios a los ciudadanos, desde el punto de vista Institucional, la OGETIC del Ministerio de Cultura busca mejoras continuas en los sistemas desplegados sin afectar su disponibilidad. Justificación Operativa: Los procesos actuales demuestran que si bien se hacen pases a producción se muestra una lentitud en el proceso y un alto riesgo, con DevOps se mitigara ello.

También la formulación del Objetivo, se plantean lo siguiente: Objetivo General: Determinar la influencia de DevOps en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura. Objetivos Específicos: O1: Determinar la influencia de DevOps en la Frecuencia de Despliegue de la Entrega Continua de la Oficina General de Estadística y Tecnología de la

Información y Comunicaciones del Ministerio de Cultura. O2: Determinar la influencia de DevOps en los Cambios Fallidos de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura. O3: Determinar la influencia de DevOps en los Cambios sin Errores de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura.

Finalmente la formulación de la hipótesis, de lo detallado anteriormente, se plantean lo siguiente: Hipótesis General: DevOps mejora la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura. Hipótesis Específicas: H1: DevOps aumenta la Frecuencia de Despliegue de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura. H2: DevOps disminuye en los Cambios Fallidos de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura. H3: DevOps aumenta los Cambios sin Errores de la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura.

II. Método

2.1 Tipo y diseño de investigación

Tipo de Investigación: Aplicada

La presente investigación es aplicada, porque se va a evaluar, así mismo comparar, con ello a interpretar para establecer precedentes y determinar casualidad a fin de obtener sus implicaciones.

Según Hernández, Fernández y Baptista (2014). La tipología aplicada es para lo que tiene adelantos como justificación así como los productos tecnológicos, generando las acciones derivadas de las investigaciones.

Diseño: Experimental

El diseño de la presente investigación es experimental, porque se influenciará intencionalmente la variable independiente (DevOps) para observar los efectos en la variable dependiente (Entrega Continua), para comprobar las hipótesis planteadas.

Según Hernández, Fernández y Baptista (2014). Mantienen que el diseño experimental viene hacer el estudio en donde intencionalmente se manipulan, ya sea una o más variables independientes las cuales son supuestas causas-antecedentes, con el fin de analizar la manipulación que genera consecuencias sobre ya sea una o más variables dependientes las cuales son los supuestos efectos-consecuentes, en ambos casos el investigador lo tiene dentro de una situación de control.

Tipología de diseño: Pre Experimental

La tipología de diseño de la presente investigación es pre experimental, la cual involucra la pre prueba que se denomina pre-test y la pos prueba que se denomina post-test, para ejecutarla en un solo grupo.

Hernández, Fernández y Baptista (2014). El diseño pre experimental es un diseño de un solo grupo, al ser caracterizado por su grado de control es bajo. Principalmente se utiliza en el inicio para un acercamiento al problema de investigación. A partir de estos se deben realizar estudios más profundos.

Para Hernández, Fernández y Baptista (2014), el diseño pre experimental tiene la clasificación de el diseño pre prueba post prueba, así como el estudio con una sola medición del caso, para la presente investigación utilizaremos el primer diseño, en donde se especifica que a un grupo se efectúa la medición de una pre prueba y post prueba.

Su representación diagramática es la siguiente:

G1 -> X -> G2

Diseños de medición de Pre-Test y Post-Test

G: La medición a los sujetos de un grupo (pre prueba, post prueba)

G1: Antes de aplicar el DevOps

G2: Después de aplicar el DevOps

X: Es la aplicación (DevOps)

2.2 Operacionalización de variables

Definición Conceptual

El presente trabajo de investigación consta de dos variables, son las siguientes:

Variable Independiente (VI): DevOps

Michael Hüttermann (2012). DevOps, un acrónimo de desarrollo y operaciones, medios para cerrar las brechas entre el desarrollo y las operaciones, por alinear incentivos y compartir enfoques de procesos y herramientas.

Variable Dependiente (VD): Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura.

Según Eder Castro Lucas (2014). La Entrega Continua viene a ser una disciplina del desarrollo de software, en donde el uso de un conjunto de patrones con las buenas prácticas, pueda lograr que el software se construya para ser rápida, frecuente, reproducible y asumiendo pocos riesgos en las puesta a producción.

Tabla 1

Identificación de variables

Variable	Descripción
Variable independiente	DevOps
Variable dependiente	Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura

Fuente: Elaboración propia

Definición Operacional

El presente trabajo de investigación consta de dos variables, son las siguientes:

Variable Independiente (VI): DevOps

DevOps, permitirá cerrar las brechas entre el desarrollo y las operaciones por alinear incentivos y compartir enfoques de procesos y herramientas.

Variable Dependiente (VD): Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura.

La Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura empleara la Entrega Continua, en donde el uso de un conjunto de patrones con las buenas prácticas, pueda lograr que el software se construya para ser rápida, frecuente, reproducible y asumiendo pocos riesgos en las puesta a producción.

Tabla 2

Operacionalización

Dimensiones	Indicadores	Ítems	Escala de medición	Niveles y rangos
Eficacia	Frecuencia de Despliegue	$FD = \frac{\text{Nro. de solicitudes}}{\text{Nro. de despliegues realizados}} * 100$	Numérico	
Eficacia	Cambios Fallidos	$CF = \frac{\text{Nro. de cambios fallidos}}{\text{Total de cambios cerrados}} * 100$	Numérico	
Eficacia	Cambios sin Errores	$CF = \frac{\text{Nro. de cambios sin errores}}{\text{Total de cambios cerrados}} * 100$	Numérico	

2.3 Población, muestra y muestreo

2.3.1 Población

Según Arias (2006). La población en donde con términos más precisos es el objetivo poblacional, se define como una agrupación finito o infinito de componentes en donde sus particularidades los cuales serán prolongado para las conclusiones de la investigación, la cual queda acotado por el problema y por los objetivos del estudio.

De acuerdo con Hernández, Fernández y Baptista (2014). La población viene a ser el agrupamiento de todas las casualidades concordantes con las especificaciones de una serie, en donde pueden ser analizados a fin de generalizar los resultados.

La población para la presente investigación es de 10 Proyectos.

2.3.2 Muestra

Para Castro (2003), la muestra está clasificada en probabilística, en donde cada uno de la totalidad de la población van a tener la misma posibilidad de conformarla, y en no probabilística, en donde no todos los miembros de la población van a tener la igualdad de posibilidad de conformarla.

De acuerdo con Hernández, Fernández y Baptista (2014). Para los procesos cuantitativos un subgrupo es la muestra de la población, de la cual se recogerán los datos, y que debe limitarse con precisión, debe representar a la población.

Según Castro (2003), Si la población viene a ser menor a 50, la muestra es igual a la población. En este sentido Ramírez (1997) establece que todas las unidades de la muestra de investigación viene a considerarse como muestra censal.

La muestra para la presente investigación es de 10 Proyectos toda vez que la muestra al ser menor o igual a 50 es igual que la población, por ende, es una muestra censal.

2.4 Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Las técnicas de recolección de datos

Según Hernández, Fernández y Baptista (2014) las técnicas de recolección de los datos tienen la posibilidad de ser múltiples, por ejemplo, los cuestionarios cerrados, registros de datos estadísticos vendrían ser en la investigación cuantitativa.

De acuerdo con Chávez (2018) las técnicas de recolección de los datos son medios que se usan para recopilar la información en una investigación, en donde pueden ser directas que son las entrevistas y las observaciones; y las indirectas que son los cuestionarios, las escalas, los inventarios y los tests.

Instrumento de recolección de datos.

Según Hernández, Fernández y Baptista (2014). Un instrumento de recolección de datos es un recurso que es utilizado por el investigador a fin de registrar toda la información como datos sobre las variables.

La ficha de registro también se le llama ficha o tarjeta de trabajo.

Según Hernández, Fernández y Baptista (2014). Su función principal es la recolección de información que se consigue en el campo al efectuar las observaciones o entrevistas. Se desarrollan con el fin de detallar un individuo u objeto. Su contenido debe ser lo más objetivo que se pueda lograr.

Validez y confiabilidad de los instrumentos.

El instrumento fue validado mediante juicio de expertos. Los expertos validaron favorablemente el instrumento siguiendo criterios de pertinencia, relevancia y claridad. Las fichas de validación se muestran en el Anexo de validez de instrumentos.

2.5 Procedimiento Método de análisis de datos

En la presente investigación se procederá con analizar los datos toda vez que la recolección metodizada así mismo los datos analizados deben de ser elegidos en función de las preguntas clave de evaluación, así como los recursos disponibles de la evaluación. Los métodos de recolección y los de análisis de datos deben seleccionarse de forma sustancial que los puntos fuertes y débiles se complementen entre sí.

2.6 Método de análisis de datos

Cuantitativo

En el presente trabajo de investigación, el método de análisis de datos es Cuantitativo, toda vez que es pre experimental y se obtienen valores que comprueben la hipótesis planteada

Según Hernández, Fernández y Baptista (2014). El método de análisis de datos es cuantitativo, toda vez que es pre experimental y las estadísticas que se consiguen puedan permitir ayudar a verificar si la hipótesis es correcta.

En esta investigación se contrasta los resultados del Pre-test que son los resultados del proceso sin aplicar el DevOps y el Post-test que son los resultados a partir de aplicar el DevOps.

Observación

Según Hernández, Fernández y Baptista (2014). La observación se refiere a la recolección de datos, el cual consiste en el registro sistemático, así mismo validado para ser confiable de comportamientos y situaciones observables, a través de un conjunto de categorías y subcategorías

2.7 Aspectos éticos

El investigador se compromete a respetar la veracidad de los resultados, la confiabilidad de los datos suministrados, la identidad de los individuos y de los objetos que participan en el estudio.

III. Resultados

La presente investigación para la determinación de la hipótesis se realizó en dos etapas, de acuerdo al diseño de estudio Pre-Experimental. En donde la primera etapa es la aplicación de la prueba pre-test, es decir que se realizó la medición de cada indicador antes de la implementación del DevOps. En donde la segunda etapa es la aplicación de la prueba post-test, es decir que se realizó nuevamente la medición de los indicadores con el DevOps implementado. De esta manera en base a los datos obtenidos en cada etapa de la investigación permite realizar las comparaciones.

En el pretest se tomó en cuenta la información de las fichas de registros de los 10 proyectos que comprende en el mes de Junio, para luego de aplicar DevOps en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, se procedió con el post-test que comprende los mismos 10 proyectos en el mes de Julio. Ver los Instrumentos.

En la presente investigación los resultados descriptivos son:

Tabla 3

Estadísticos descriptivos de la Frecuencia de despliegue

	N	Media	Desv. Desviacion	Desv. Error Promedio
Pre Test de Frecuencia de Despliegue	15	32,0100	11,83176	3,05494
Post Test de Frecuencia de Despliegue	15	81,2700	9,25048	2,38846

En la tabla 3 denotamos que la frecuencia de despliegue de Pre test existe una media de 32,0100 con una desv. desviación de 11,83176 y desv. error promedio de 3,05494 , mientras que en el Post test existe una media de 81,2700 con una desv. desviación de 9,25048 y desv. error promedio de 2,38846.

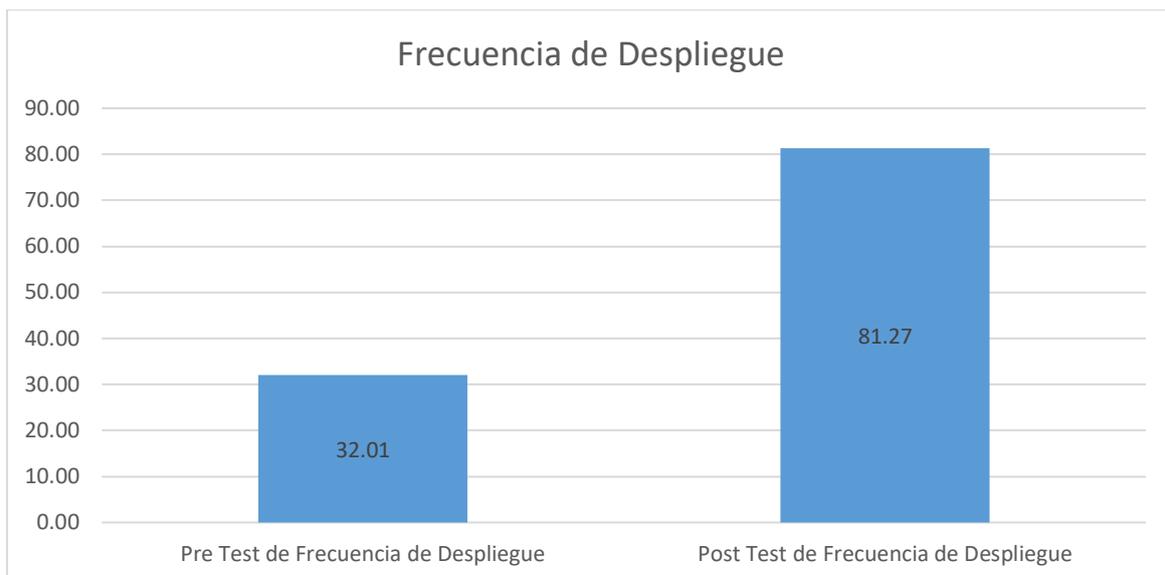


Figura 1: Media de Frecuencia de Despliegue

Tabla 4

Estadísticos descriptivos de los Cambios Fallidos

	N	Media	Desv. Desviacion	Desv. Error Promedio
Pre Test de Cambios Fallidos	15	84,4453	18,05614	4,66208
Post Test de Cambios Fallidos	15	27,0240	11,18910	2,88901

En la tabla 4 denotamos que los cambios fallidos de Pre test existe una media de 84,4453 con una desv. desviación de 18,05614 y desv. error promedio de 4,66208 , mientras que en el Post test existe una media de 27,0240 con una desv. desviación de 11,18910 y desv. error promedio de 2,88901.

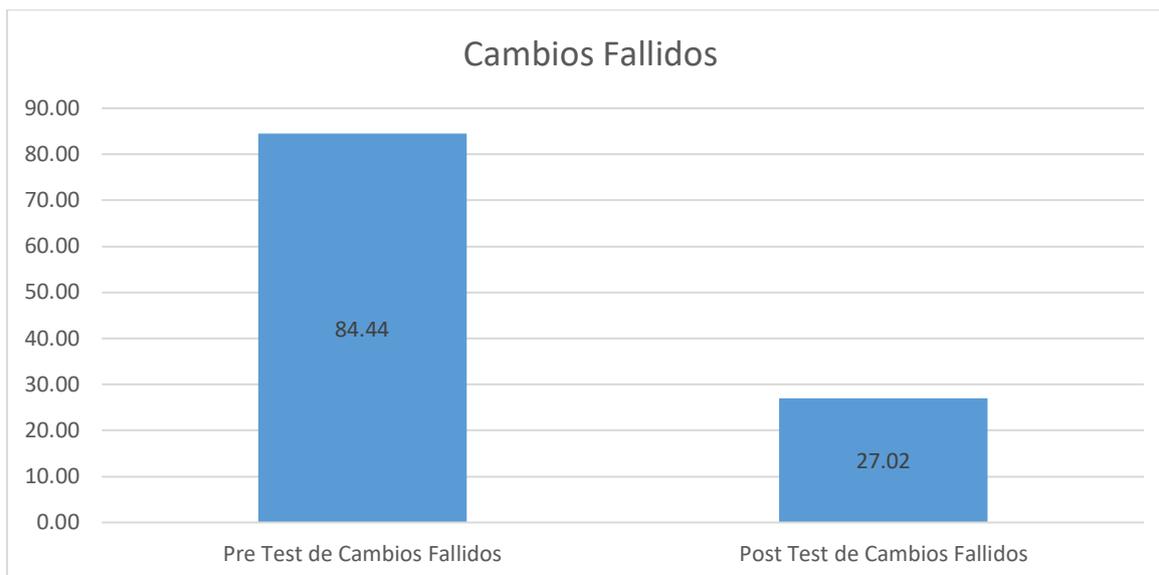


Figura 2: Media de Cambios Fallidos

Tabla 5

Estadísticos descriptivos de los Cambios sin Errores

	N	Media	Desv. Desviacion	Desv. Error Promedio
Pre Test de Cambios sin Errores	15	58,3327	24,60102	6,35196
Post Test de Cambios sin Errores	15	83,9273	10,57315	2,72998

Fuente: Elaboración propia

En la tabla 5 denotamos que los cambios sin errores de Pre test existe una media de 58,3327 con una desv. desviación de 24,60102 y desv. error promedio de 6,35196 , mientras que en el Post test existe una media de 83,9273 con una desv. desviación de 10,57315 y desv. error promedio de 2,72998.

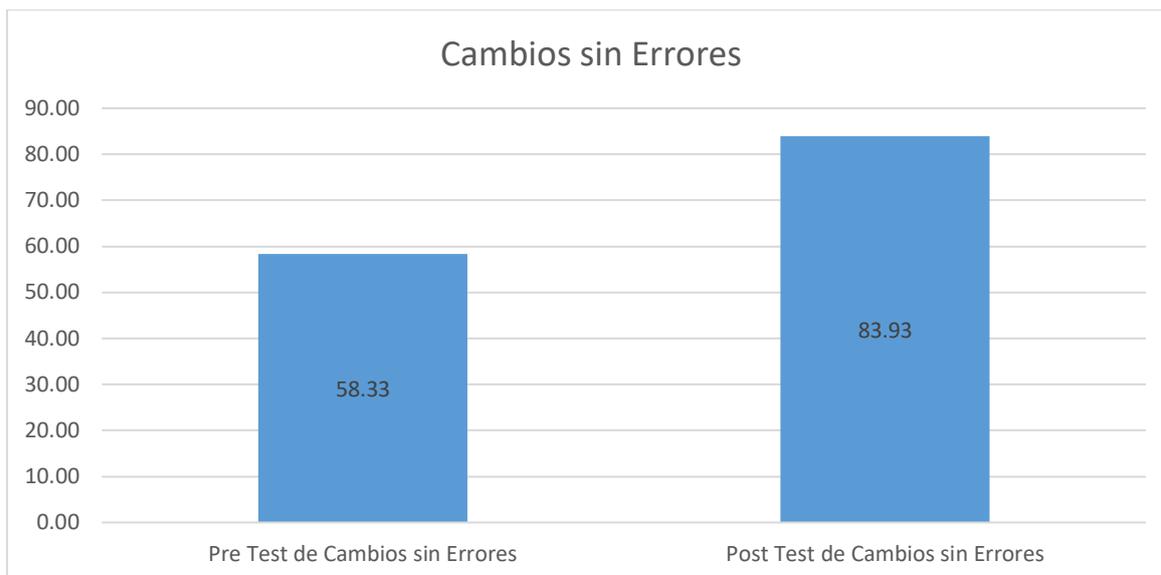


Figura 3: Media de Cambios sin Errores

En la presente investigación los resultados de la normalidad son:

Frecuencia de Despliegue:

Como el tamaño de la muestra es de 15 días, se usó la prueba de Shapiro-Wilk. En la tabla 6 se visualiza las pruebas de normalidad para el indicador Frecuencia de Despliegue en PreTest y PostTest

Tabla 6

Prueba de normalidad de la Frecuencia de despliegue

	Kolmogorov-Smirnova			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
FD_PRE	0.143	15	,200*	0.961	15	0.713
FD_POST	0.218	15	0.054	0.915	15	0.160

***. Esto es un límite inferior de la significación verdadera.**

a. Corrección de significación de Lilliefors

Como se muestra en la tabla 6 el valor Sig para PreTest es de 0.713 y para el PostTest es de 0.160, debido a que el PreTest y PostTest son mayores a 0.05 por lo tanto es una distribución normal.

Cambios Fallidos:

Como el tamaño de la muestra es de 15 días, se usó la prueba de Shapiro-Wilk. En la tabla 7 se visualiza las pruebas de normalidad para el indicador Cambios Fallidos en PreTest y PostTest

Tabla 7

Prueba de normalidad de los Cambios Fallidos

Pruebas de normalidad						
	Kolmogorov-Smirnova			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
CF_PRE	0.339	15	0.000	0.766	15	0.001
CF_POST	0.156	15	,200*	0.930	15	0.268

***. Esto es un límite inferior de la significación verdadera.**

a. Corrección de significación de Lilliefors

Como se muestra en la tabla 7 el valor Sig para PreTest es de 0.713 y para el PostTest es de 0.160, debido a que el PreTest y PostTest son mayores a 0.05 por lo tanto es una distribución normal.

Cambios sin Errores:

Como el tamaño de la muestra es de 15 días, se usó la prueba de Shapiro-Wilk. En la tabla 8 se visualiza las pruebas de normalidad para el indicador Cambios sin Errores en PreTest y PostTest

Tabla 8

Prueba de normalidad de los Cambios sin Errores

Pruebas de normalidad						
	Kolmogorov-Smirnova			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
CSE_PRE	0.299	15	0.001	0.815	15	0.006
CSE_POST	0.211	15	0.072	0.893	15	0.075

a. Corrección de significación de Lilliefors

Como se muestra en la tabla 8 el valor Sig para PreTest es de 0.006 y para el PostTest es de 0.075, debido a que el PreTest y PostTest son mayores a 0.05 por lo tanto es una distribución normal.

En la presente investigación las Hipótesis Estadística es:

Frecuencia de Despliegue:

H1: DevOps influye en el aumento de la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura

H0: DevOps no influye en el aumento de la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura

Tabla 9

Estadístico de Prueba T de Frecuencia de Despliegue

Prueba para una muestra						
Valor de prueba = 0						
	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
FD_PRE	10.478	14	0.000	32.01000	25.4578	38.5622
FD_POST	34.026	14	0.000	81.27000	76.1473	86.3927

Denotamos en la tabla 9 que la significancia es de 0.000 por lo que afirmamos que DevOps influye en el aumento de la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura

Cambios Fallidos:

H1: DevOps influye en la disminución de los Cambios Fallidos de la Entrega Continua del Ministerio de Cultura

H0: DevOps no influye en la disminución de los Cambios Fallidos de la Entrega Continua del Ministerio de Cultura

Tabla 10

Estadístico de Prueba T de Frecuencia de Despliegue

Prueba para una muestra						
	Valor de prueba = 0				95% de intervalo de confianza de la diferencia	
	t	gl	Sig. (bilateral)	Diferencia de medias	Inferior	Superior
CF_PRE	18.113	14	0.000	84.44533	74.4462	94.4445
CF_POST	9.354	14	0.000	27.02400	20.8277	33.2203

Denotamos en la tabla 10 que la significancia es de 0.000 por lo que afirmamos que DevOps influye en la disminución de los Cambios Fallidos de la Entrega Continua del Ministerio de Cultura

Cambios sin Errores:

H1: DevOps influye en el aumento de los Cambios sin Errores de la Entrega Continua del Ministerio de Cultura

H0: DevOps no influye en el aumento de los Cambios sin Errores de la Entrega Continua del Ministerio de Cultura

Tabla 11

Estadístico de Prueba T de Frecuencia de Despliegue

Prueba para una muestra						
	Valor de prueba = 0				95% de intervalo de confianza de la diferencia	
	t	gl	Sig. (bilateral)	Diferencia de medias	Inferior	Superior
CSE_PRE	9.183	14	0.000	58.33267	44.7091	71.9563
CSE_POST	30.743	14	0.000	83.92733	78.0721	89.7826

Denotamos en la tabla 11 que la significancia es de 0.000 por lo que afirmamos que DevOps influye en el aumento de los Cambios sin Errores de la Entrega Continua del Ministerio de Cultura

En la Hipótesis General

H1: DevOps influye en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura.

H0: DevOps no influye en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura.

Tabla 12

Hipotesis General

Prueba para una muestra						
Valor de prueba = 0						
	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
FD_PRE	10.478	14	0.000	32.01000	25.4578	38.5622
FD_POST	34.026	14	0.000	81.27000	76.1473	86.3927
CF_PRE	18.113	14	0.000	84.44533	74.4462	94.4445
CF_POST	9.354	14	0.000	27.02400	20.8277	33.2203
CSE_PRE	9.183	14	0.000	58.33267	44.7091	71.9563
CSE_POST	30.743	14	0.000	83.92733	78.0721	89.7826

Denotamos en la tabla 12 que la significancia es de 0.000 por lo que afirmamos que DevOps influye en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura

IV. Discusión

En el año 2018, Luis Alberto Iñiguez Sánchez en tesis de la maestría denominada “Arquitectura para la entrega continua de software con despliegue en contenedores”, propone una única arquitectura para el software a desarrollar, cada vez que los despliegues de versiones nuevas en las aplicaciones sean extensas las tareas además de complicadas, en donde se introduce los errores con facilidad de regresión en el software de producción, con ello dificultando que la calidad sea asegurada. Con ello implementa la entrega continua de software, de esta manera el equipo será potenciado para eliminar los riesgos inherentes a cada despliegue de producto.

Comparando el aumento de la frecuencia de despliegue, la disminución en los cambios fallidos y aumentando los cambios sin errores de la entrega continua de la presente investigación con lo presentado por Luis Alberto Iñiguez Sánchez (2018), el cual logra influir positivamente con su también entrega continua permitiendo como resultado final obtener un modelo para la entrega continua que está basado en la tecnología que mitiga los riesgos inherentes en cada despliegue toda vez que cuenta con un soporte completo de stack tecnológico.

La virtualización de recursos, es uno de los conceptos esenciales de la computación en la nube, refiriéndose a la creación de una versión de software en vez de una, incluyendo y no limitado a la plataforma de hardware, con su sistema operativo y su conjunto de almacenamiento o recursos de la red informática.

La virtualización utiliza la capa intermedia en el software, que esta sobre un sistema el cual permite proporcionar abstracción de los recursos virtuales multiples. En la actualidad se pueden considerar dos grandes grupos en los modelos de virtualización, vendrían a ser las máquinas virtuales y los contenedores.

Las maquinas virtuales en donde son conocidos como VM y son los recursos virtuales son componentes de software, que podrían ser detallados como ambientes de ejecución aislados.

Según Singh & Singh (2016). La virtualización en el mercado existen diferentes técnicas, una de las más folclóricas es la basada en hypervisor, que necesita de un Virtual Machine Monitor (VMM) el cual se ejecuta sobre el anfitrión del sistema operativo, así proporciona una abstracción completa, permitiendo que en un solo anfitrión real se pueda ejecutar múltiples sistemas operativos, por ejemplo las herramientas como Xen26, KVM27, Virtual Box28 y WM-Ware29.

Los contenedores, según Celesti, Mulfari, Fazio, Villari, & Puliafito, (2016), es la técnica de virtualización de sistema operativo, permitiendo la ejecución de diferentes máquinas virtuales sobre el kernel del anfitrión sistema operativo.

Los contenedores en el sistema operativo comparten un solo kernel, por ende la virtualización basada en contenedores a comparación de la basada en hypervisor, tiene un aislamiento más frágil según Celesti, Mulfari, Fazio, Villari, & Puliafito (2016).

Desde el punto de vista de los usuarios, es imperceptible porque se ve y ejecuta exactamente como un sistema operativo independiente cada contenedor. En consecuencia, permite un mayor número de despliegues de contenedores que máquinas virtuales en un mismo anfitrión físico.

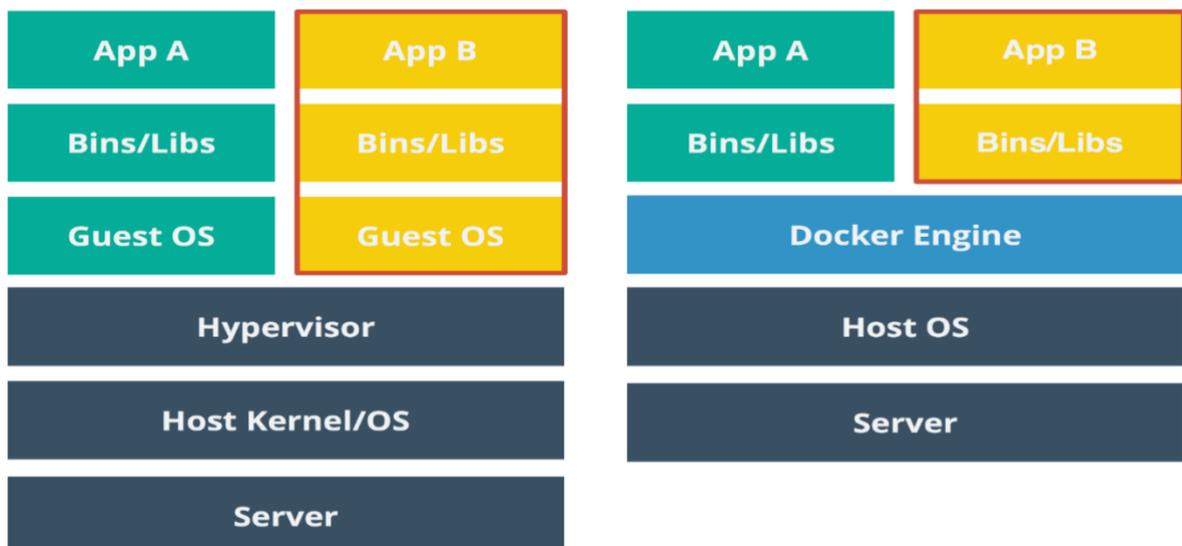


Figura 4: Máquina Virtual y Contenedor

La Figura 4 muestra la diferencia puntual entre las tecnologías de virtualización. En donde la virtualización basada en hypervisor permite la abstracción para sistemas operativos invitados completos que son uno por VM, la virtualización basada en contenedores funciona a nivel del sistema operativo proporcionando abstracciones directamente para los procesos de invitado según Li, Kihl, Lu, & Andersson (2017)

En la presente investigación se implementó el contenedor Docker, porque toda vez que las soluciones de hypervisor funcionan como abstracción a nivel de hardware, en cambio la virtualización basada en contenedores opera a nivel de llamada del sistema proporcionando mejores beneficios.

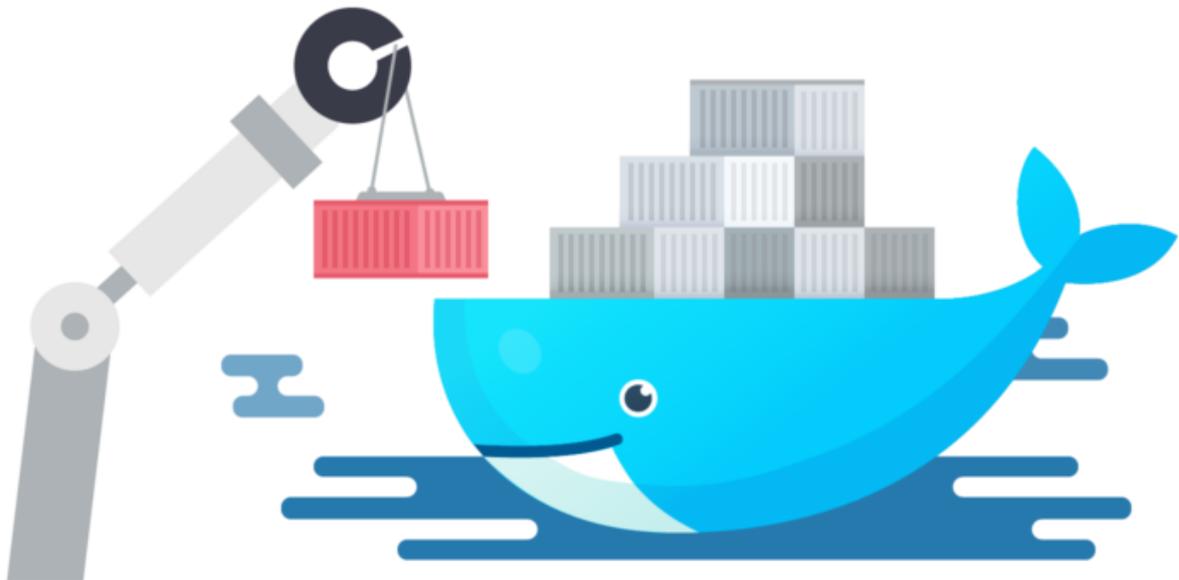


Figura 5: Docker que es un contenedor

Con ello en relación a lo indicado por Luis Alberto Iñiguez Sánchez (2018), se usa contenedores en Docker en donde ofrece mejoras en la entrega continua al tener por aplicaciones un contenedor dedicado, así un mismo servidor físico puede albergar varios contenedores con diferentes recursos.

Así mismo se implementa el sistema de versión de controles. Según Just, Herzig, Czerwonka, & Murphy (2016). Los sistemas de control de versiones modernos vienen implementados mediante una arquitectura distribuida, de esta manera facilitando el uso de las metodologías ágiles en los equipos del desarrollo de software, mediante sus procesos de integración cada vez más rápidos, a si mismo mejora el aislamiento para garantizar la independencia de los cambios entre los desarrolladores.

En la presente investigación se implementó el sistema de control de versiones GitLab, el cual cuenta con la fuente de los proyectos web, de esta manera los cambios realizados se pueden verificar que línea se ha cambiado, actualizado, eliminado, agregado.



Figura 6: GitLab que es un sistema de versión de contenidos

Con ello en relación a lo indicado por Luis Alberto Iñiguez Sánchez (2018), se usa el sistema de control de versiones GitLab en donde ofrece mejoras en la entrega continua al tener todo los registros de los codigos y poder realizar los pases a producción de una manera transparente.

Adicional se está usando a Composer, que es un administrador de paquetes del lenguaje de programación de PHP, el cual es a nivel de aplicación, donde proporciona un formato estándar el cual permite la administración de las dependencias del software PHP y de los requerimientos de las bibliotecas.

```
Composer version 1.8.3 2019-01-30 08:31:33

Usage:
  command [options] [arguments]
```

Figura 7: Composer que es un administrador de paquetes

También se usa Drush que es una aplicación basada en shell de software de computadora utilizada para controlar, manipular y administrar sitios web de Drupal. En la superficie, drush es una herramienta para actualizar los módulos del sitio, sin embargo, Drush tiene una lista más completa de características

```
Drush Commandline Tool 9.5.2

Run `drush help [command]` to view command-specific help. Run `drush topic` to read even more documentation.

Available commands:
_global:
  help          Display usage details for a command.
  list          List available commands.
  browse        Display a link to a given path or open link in a browser.
  version       Show Drush version.
  runserver (rs, serve)  Runs PHP's built-in http server for development.
  updatedb (updb)  Apply any database updates required (as with running update.php).
  generate (gen)  Generate boilerplate code for modules/plugins/services etc.
  config:pull (cpull)  Export and transfer config from one environment to another.
  drupal:directory (dd)  Return the filesystem path for modules/themes and other key folders.
  user:login (uli)  Display a one time login link for user ID 1, or another user.
  entity:updates (entup)  Apply pending entity schema updates.
  updatedb:status (updbst)  List any pending database updates.
  pm:security (sec)  Check Drupal Composer packages for pending security updates.
```

Figura 8: Drush administra por consola a Drupal

Para los proyectos web se usa Drupal, que si bien es un sistema de gestión de contenidos o CMS libre, también calza como un Framework toda vez que es modular, multipropósito y muy configurable, lo cual permite publicar artículos con imágenes y archivos, así también ofrece la posibilidad de otros servicios añadidos, entre los cuales son los blogs, foros, votaciones, encuestas, administración de usuarios y permisos, así mismo conectarse y operar con sistemas de información y aplicaciones móviles mediante servicios.



Figura 9: Drupa que es un Framework toda vez que es modular, multipropósito y muy configurable

En relación a lo adicional y a lo indicado por Luis Alberto Iñiguez Sánchez (2018) que si bien usa alternativas similares de acuerdo a su trabajo de investigación, estas son también efectivas para poder realizar la entrega continua.

Finalmente, los resultados confirman la hipótesis planteada. Ya que se logró el aumento de la frecuencia de despliegue, la disminución en los cambios fallidos y aumentando los cambios sin errores de la entrega continua mediante el DevOps.

V. Conclusiones

La conclusión general es que DevOps influye positivamente en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, donde con DevOps se logró aumentar la frecuencia de despliegue en un 49.26%, disminuir los cambios fallidos en un 57.42% y aumentar los cambios sin errores en un 25.6%. en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura

En donde a detalle las conclusiones:

Primera: DevOps aumenta en un 49.26% la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura

Segunda: DevOps disminuye en un 57.42% los Cambios Fallidos de la Entrega Continua del Ministerio de Cultura

Tercera: DevOps aumenta en un 25.6% los Cambios sin Errores de la Entrega Continua del Ministerio de Cultura.

VI. Recomendaciones

Toda vez que DevOps influye positivamente en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, se recomienda:

Primera. Los proyectos web están beneficiados al usar DevOps, se recomienda abarcar a otros proyectos para poder obtener los beneficios demostrados.

Segunda. Cuando se creen proyectos nuevos, se recomienda iniciar con DevOps así el pase a producción tenderá a ser transparente y sin dificultades.

Tercera. Los componentes usados se recomiendan mantener actualizados para no perder las nuevas características que se van entregando a lo largo del tiempo.

VII. Propuesta

Para continuar con DevOps que influye positivamente en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, se propone:

Primero. Automatizar las actualizaciones de los componentes usados para mantenerse al día con los cambios lanzados.

Segundo. Integrar los componentes para un mejor desempeño en conjunto así poder tener toda una línea de despliegues.

Tercero. Seguir innovando en la estructura diseñada de acuerdo a los siguientes avances tecnológicos que van a surgir en el camino.

Referencias

- Arias, F. (2006). *El proyecto de investigación: Introducción a la metodología científica*. (5ª ed.) Editorial Episteme, Venezuela
- Bass, L. (2017). The Software Architect and DevOps. *IEEE Software*, 35(1), 8-10. <https://doi.org/10.1109/MS.2017.4541051>
- Beck, K. (February de 2001). *Agile Manifest Principles*. Obtenido de <http://agilemanifesto.org/iso/es/principles.html>
- Bellomo, S., Ernst, N., Nord, R., & Kazman, R. (2014). *Toward Design Decisions to Enable Deployability*. 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (págs. 702 - 707). Atlanta, USA : IEEE.
- Benson, J., Prevost, J., & Rad , P. (2016). *Survey of Automated Software Deployment for Computational and Engineering Research*. *Systems Conference (SysCon)*, 2016 Annual IEEE. Orlando, USA: IEEE.
- Castro, M. (2003). *El proyecto de investigación y su esquema de elaboración*. (2ª.ed.) Editorial Uyapar, Caracas
- Celesti, A., Mulfari, D., Fazio, M., Villari, M., & Puliafito, A. (2016). *Exploring Container Virtualization in IoT Clouds*. 2016 IEEE International Conference on Smart Computing (SMARTCOMP) (págs. 1-6). St. Louis, USA : IEEE.
- Chen, L. (2015). *Continuous Delivery: Huge Benefits, but Challenges Too*. *IEEE Software*, 50-54. USA : IEEE.
- Cito, J., Schermann, G., Wittern, J. E., Leitner, P., Zumberi, S., & Gall, H. C. (2017). *An Empirical Analysis of the Docker Container Ecosystem on GitHub*. 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR) (págs. 323 - 333). Buenos Aires, Argentina : IEEE.

- Dustin, E., Garrett, T., & Gauf, B. (2009). *Implementing automated software testing : how to save time and lower costs while raising quality*. Pearson Education, Addison-Wesley.
- Duvall, P., Matyas, S., & Glover, A. (2007). *Continuous Integration - Improving Software Quality and Reducing Risk*. Pearson Education, Addison-Wesley Professional.
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). *DevOps*. IEEE Software, 33(3), 94-100. <https://doi.org/10.1109/MS.2016.68>
- Erich, F., Amrit, C., & Daneva, M. (2014). *Cooperation between information system development and operations* (pp. 1-1). Association for Computing Machinery (ACM). <https://doi.org/10.1145/2652524.2652598>
- Fitzgerald, B., & Stol, K.-J. (2015). *Continuous software engineering: A roadmap and agenda*. The Journal of Systems and Software, USA
- Fowler, M. (1 de Mayo de 2006). Continuous Integration. <https://martinfowler.com/articles/continuousIntegration.html>
- Fowler, M. (Mayo de 2013). Continuous Delivery. <http://martinfowler.com/bliki/ContinuousDelivery.html>
- Gomede, & Barros, R. (2015). *A Practical Approach to Software Continuous Delivery*. ICONS 2015 : The Tenth International Conference on Systems (págs. 98-101). SEKE.
- Hernández, R., Fernández, C. y Baptista, P. (2014). *Metodología de la investigación* (6ª Edición). McGraw-Hill / Interamericana Editores, S.A. Mexico.

- Humble, J. (13 de 08 de 2010). Continuous Delivery vs Continuous Deployment. Obtenido de <https://continuousdelivery.com/2010/08/continuous-delivery-vscontinuous-deployment/>
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison Wesley. USA
- Hussaini, S. W. (2014). *Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach*. In 2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014 (pp. 178-183). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ITSC.2014.6957687>
- Hüttermann, M. (2012). *DevOps for Developers*. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4302-4570-4>
- Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). *What is DevOps? In Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops* (pp. 1-11). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2962695.2962707>
- Jaramillo, D. (2016). *Leveraging microservices architecture by using Docker technology*. SoutheastCon 2016. Norfolk: IEEE.
- Just, S., Herzig, K., Czerwonka, J., & Murphy, B. (2016). *Switching to Git: the Good, the Bad, and the Ugly*. IEEE 27th International Symposium on Software Reliability Engineering (págs. 400-411). Ottawa,Canada : IEEE.
- Kang, H., Le, M., & Tao, S. (2016). *Container and Microservice Driven Design for Cloud Infrastructure DevOps*. IEEE International Conference on Cloud Engineering. Berlín: IEEE computer society.

- Koc, A., & Uz, A. (2015). A Survey of Version Control Systems. Obtenido de https://www.researchgate.net/publication/266866968_A_Survey_of_Version_Control_Systems
- Kumar, K., & Kurhekar, M. (2016). *Economically Efficient Virtualization Over Cloud Using Docker Containers*. 2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) (págs. 95 - 100). Bangalore, India : IEEE.
- Li, Z., Kihl, M., Lu, Q., & Andersson, J. A. (2017). *Performance Overhead Comparison between Hypervisor and Container based Virtualization*. 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA) (págs. 955 - 962). Taipei, Taiwan: IEEE.
- Luis Alberto Iñiguez Sánchez (2018). *Arquitectura para la entrega continua de software con despliegue en contenedores*, Universidad de Cuenca, Cuenca
- Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). *Dimensions of devOps*. In *Lecture Notes in Business Information Processing* (Vol. 212, pp. 212-217). Springer Verlag. https://doi.org/10.1007/978-3-319-18612-2_19
- Mårtensson, T., Ståhl, D., & Bosch, J. (2017). *Continuous Integration Impediments in Large-Scale Industry Projects*. 2017 IEEE International Conference on Software Architecture (ICSA) (págs. 169 - 178). Gothenburg, Sweden: IEEE.
- Meyer, S., Healy, P., Lynn, T., & Morrison, J. (2013). *Quality Assurance for Open Source Software Configuration Management. Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2013 15th International Symposium on. Timisoara, Romania : IEEE.
- Michelsen, J. (2014). *A Pragmatic Guide to Getting Started with DevOps*. Obtenido de <https://www.ca.com/content/dam/ca/us/files/ebook/a-pragmatic-guide-to-getting-started-with-devops.pdf>

- Nielsen, P. A., Winkler, T. J., & Nørbjerg, J. (2017). *Closing the IT development-operations gap: The devops knowledge sharing framework*. In CEUR Workshop Proceedings (Vol. 1898). CEUR-WS.
- Pulkkinen, V. (2013). *Continuous Deployment of Software*. Cloud-Based Software Engineering - Proceedings of the seminar N°. 58312107, (págs. 46-52). Helsinki, Finlandia.
- Pulkkinen, V. (2016). *Synthesizing Perceived Challenges in Continuous Delivery - A Systematic Literature Review*. Master Science Tesis, 46 - 52. Helsinki, Finlandia.
- Ramírez, T. (1997). *Cómo hacer un proyecto de investigación*. Editorial Panapo, Caracas.
- Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). *DevOps for Digital Leaders*. DevOps for Digital Leaders. Apress. <https://doi.org/10.1007/978-1-4842-1842-6>
- Sanchez, U., Reyes, C., Mejia, K., (2018). *Manual de términos en investigación científica, tecnológica y humanística*. Universidad Ricardo Palma, Perú.
- Shahin, M., Ali Babar, M., & Zhu, L. (2017). *Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices*. IEEE Access. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2017.2685629>
- Shahin, M., Babar, A., & Zhu, L. (2017). *Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices*. IEEE Software.
- Shahin, M., Zahedi, M., Babar, M. A., & Zhu, L. (2017). *Adopting Continuous Delivery and Deployment* (pp. 384-393). Association for Computing Machinery (ACM). <https://doi.org/10.1145/3084226.3084263>

- Sharma, S., & Coyne, B. (2015). *DevOps For Dummies*. John Wiley & Sons, Inc. (Vol. 1, p. 76). <https://doi.org/10.1017/CBO9781107415324.004>
- Singh, S., & Singh, N. (2016). *Containers & Docker: Emerging Roles & Future of Cloud Technology*. 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (págs. 804 - 807). Bangalore, India: IEEE.
- Ståhl, D., & Bosch, J. (2014). *Modeling continuous integration practice differences in industry software development*. *The Journal of Systems and Software*, 48-59.
- Stolberg, S. (2009). *Enabling Agile Testing through Continuous Integration*. Agile Conference, AGILE '09. Chicago, USA: IEEE.
- Sultanía, A. (2015). *Developing Software Product and Test Automation Software Using Agile Methodology*. Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT) (págs. 1 - 4). Hooghly, India: IEEE.
- Virmani, M. (2015). *Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery*. Fifth international conference on Innovative Computing Technology (págs. 78-82). Pontevedra, Spain: IEEE.
- Wang, W., Casale, G., & Iuhasz, G. (2017). *Closing the loop between Ops and Dev*. In *SpringerBriefs in Applied Sciences and Technology* (pp. 95-105). Springer Verlag. https://doi.org/10.1007/978-3-319-46031-4_10
- Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). *DevOps and Its Practices*. *IEEE Software*, 33(3), 32-34. <https://doi.org/10.1109/MS.2016.81>

Anexos

Matriz de consistencia

Título: DevOps en la Entrega Continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura

Autor: Jhonny Peralta Mori

Problema	Objetivos	Hipótesis	Variables e indicadores				
<p>Problema General: ¿De qué manera influye el DevOps en la Entrega Continua del Ministerio de Cultura?</p> <p>Problemas Específicos: P1 ¿En qué medida el uso de DevOps influye en la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura?</p> <p>P2 ¿En qué medida el uso de DevOps influye en los Cambios Fallidos de la Entrega Continua del Ministerio de Cultura?</p> <p>P3 ¿En qué medida el uso de DevOps influye en los Cambios sin Errores en la Entrega Continua del Ministerio de Cultura?</p>	<p>Objetivo general: Determinar la influencia de DevOps en la Entrega Continua del Ministerio de Cultura.</p> <p>Objetivos específicos: O1: Determinar la influencia de DevOps en la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura.</p> <p>O2: Determinar la influencia de DevOps en los Cambios Fallidos de la Entrega Continua en el Ministerio de Cultura.</p> <p>O3: Determinar la influencia de DevOps en los Cambios sin Errores de la Entrega Continua del Ministerio de Cultura.</p>	<p>Hipótesis general: DevOps mejora en la Entrega Continua del Ministerio de Cultura.</p> <p>Hipótesis específicas: H1: DevOps aumenta la Frecuencia de Despliegue de la Entrega Continua del Ministerio de Cultura.</p> <p>H2: DevOps disminuye en los Cambios Fallidos de la Entrega Continua del Ministerio de Cultura.</p> <p>H3: DevOps aumenta los Cambios sin Errores de la Entrega Continua del Ministerio de Cultura</p>	Variable 1: DevOps				
			Dimensiones	Indicadores	Ítems	Escala de medición	Niveles y rangos
			Variable 2: Entrega Continua				
			Dimensiones	Indicadores	Ítems	Escala de medición	Niveles y rangos
			Eficacia	Frecuencia de Despliegue	$FD = \frac{Nro. de despliegues solicitados}{Nro. de despliegues realizados} * 100$	Numérico	
Eficacia	Cambios Fallidos	$CF = \frac{Nro. de cambios fallidos}{Total de cambios cerrados} * 100$	Numérico				
Eficacia	Cambios sin Errores	$CSE = \frac{Nro. de cambios sin errores}{Total de cambios cerrados} * 100$	Numérico				

Diseño de investigación	Población y muestra	Técnicas e instrumentos	Estadística a utilizar
<p>Diseño: Pre Experimental Hernández, Fernández y Baptista (2014). El diseño pre experimental es un diseño de un solo grupo, se caracterizan porque su grado de control es bajo. Principalmente se utiliza como un acercamiento inicial al problema de investigación. A partir de estos se deben realizar estudios más profundos.</p> <p>Método: Cuantitativo Según Hernández, Fernández y Baptista (2014). El método de análisis de datos es cuantitativo, ya que es pre experimental y se obtienen estadísticas que ayuden a comprobar si la hipótesis es correcta.</p>	<p>Población: 10 proyectos</p> <p>Tipo de muestreo: Censal Ramírez (1997) establece que la muestra censal es aquella donde todas las unidades de investigación son consideradas como muestra</p> <p>Tamaño de muestra: 10 proyectos</p>	<p>Variable 1: DevOps Técnicas: Fichaje Instrumentos: Ficha de registro Autor: Año: Monitoreo: Ámbito de Aplicación: Forma de Administración:</p> <p>Variable 2: Entrega Continua Técnicas: Fichaje Instrumentos: Ficha de registro Autor: Año: Monitoreo: Ámbito de Aplicación: Forma de Administración:</p>	<p>DESCRIPTIVA:</p> <p>INFERENCIAL:</p>

Instrumentos

FICHA DE REGISTRO PARA EL INDICADOR: “FRECUENCIA DE DESPLIEGUE” (PRE-TEST)

INVESTIGADOR	JHONNY PERALTA MORI
EMPRESA	MINISTERIO DE CULTURA
AREA	OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES
FECHA INICIO	03/06/2019
FECHA FIN	21/06/2019

$$FD = \frac{\text{Nro. de despliegues solicitados}}{\text{Nro. de despliegues realizados}} * 100$$

FECHA	AREA	Nº DESPLIEGUES REALIZADOS	Nº DESPLIEGUES SOLICITADOS	FRECUEN- CIA DE DESPLIEGUE
3/06/19	OGETIC	3	9	33.33
4/06/19	OGETIC	1	7	14.29
5/06/19	OGETIC	2	8	25.00
6/06/19	OGETIC	3	7	42.86
7/06/19	OGETIC	1	8	12.50
10/06/19	OGETIC	2	7	28.57
11/06/19	OGETIC	3	9	33.33
12/06/19	OGETIC	2	8	25.00
13/06/19	OGETIC	4	7	57.14
14/06/19	OGETIC	3	8	37.50
17/06/19	OGETIC	2	8	25.00
18/06/19	OGETIC	2	6	33.33
19/06/19	OGETIC	2	8	25.00
20/06/19	OGETIC	3	7	42.86
21/06/19	OGETIC	4	9	44.44

**FICHA DE REGISTRO PARA EL INDICADOR:
“CAMBIOS FALLIDOS” (PRE-TEST)**

INVESTIGADOR	JHONNY PERALTA MORI
EMPRESA	MINISTERIO DE CULTURA
AREA	OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES
FECHA INICIO	03/06/2019
FECHA FIN	21/06/2019

$$CF = \frac{\text{Nro. de cambios fallidos}}{\text{Total de cambios cerrados}} * 100$$

FECHA	AREA	Nº CAMBIOS FALLIDOS	TOTAL CAMBIOS CERRADOS	CAMBIOS FALLIDOS
3/06/19	OGETIC	2	3	66.67
4/06/19	OGETIC	1	1	100.00
5/06/19	OGETIC	1	2	50.00
6/06/19	OGETIC	2	3	66.67
7/06/19	OGETIC	1	1	100.00
10/06/19	OGETIC	2	2	100.00
11/06/19	OGETIC	2	3	66.67
12/06/19	OGETIC	2	2	100.00
13/06/19	OGETIC	3	4	75.00
14/06/19	OGETIC	2	3	66.67
17/06/19	OGETIC	2	2	100.00
18/06/19	OGETIC	2	2	100.00
19/06/19	OGETIC	2	2	100.00
20/06/19	OGETIC	3	3	100.00
21/06/19	OGETIC	3	4	75.00

**FICHA DE REGISTRO PARA EL INDICADOR:
“CAMBIOS SIN ERRORES” (PRE-TEST)**

INVESTIGADOR	JHONNY PERALTA MORI
EMPRESA	MINISTERIO DE CULTURA
AREA	OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES
FECHA INICIO	01/07/2019
FECHA FIN	19/07/2019

$$CSE = \frac{\text{Nro. de cambios sin errores}}{\text{Total de cambios cerrados}} * 100$$

FECHA	AREA	N° CAMBIOS SIN ERRORES	TOTAL CAMBIOS CERRADOS	CAMBIOS SIN ERRORES
3/06/19	OGETIC	1	3	33.33
4/06/19	OGETIC	1	1	100.00
5/06/19	OGETIC	1	2	50.00
6/06/19	OGETIC	2	3	66.67
7/06/19	OGETIC	1	1	100.00
10/06/19	OGETIC	1	2	50.00
11/06/19	OGETIC	1	3	33.33
12/06/19	OGETIC	1	2	50.00
13/06/19	OGETIC	2	4	50.00
14/06/19	OGETIC	1	3	33.33
17/06/19	OGETIC	2	2	100.00
18/06/19	OGETIC	1	2	50.00
19/06/19	OGETIC	1	2	50.00
20/06/19	OGETIC	1	3	33.33
21/06/19	OGETIC	3	4	75.00

**FICHA DE REGISTRO PARA EL INDICADOR:
“FRECUENCIA DE DESPLIEGUE” (POST-TEST)**

INVESTIGADOR	JHONNY PERALTA MORI
EMPRESA	MINISTERIO DE CULTURA
AREA	OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES
FECHA INICIO	01/07/2019
FECHA FIN	19/07/2019

$$FD = \frac{\text{Nro. de despliegues solicitados}}{\text{Nro. de despliegues realizados}} * 100$$

FECHA	AREA	N° DESPLIEGUES REALIZADOS	N° DESPLIEGUES SOLICITADOS	FRECUEN- CIA DE DESPLIEGUE
1/07/19	OGETIC	7	8	87.50
2/07/19	OGETIC	5	7	71.43
3/07/19	OGETIC	6	7	85.71
4/07/19	OGETIC	8	9	88.89
5/07/19	OGETIC	5	7	71.43
8/07/19	OGETIC	6	8	75.00
9/07/19	OGETIC	7	8	87.50
10/07/19	OGETIC	6	9	66.67
11/07/19	OGETIC	7	7	100.00
12/07/19	OGETIC	6	8	75.00
15/07/19	OGETIC	5	7	71.43
16/07/19	OGETIC	7	8	87.50
17/07/19	OGETIC	6	7	85.71
18/07/19	OGETIC	7	9	77.78
19/07/19	OGETIC	7	8	87.50

**FICHA DE REGISTRO PARA EL INDICADOR:
“CAMBIOS FALLIDOS” (POST-TEST)**

INVESTIGADOR	JHONNY PERALTA MORI
EMPRESA	MINISTERIO DE CULTURA
AREA	OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES
FECHA INICIO	01/07/2019
FECHA FIN	19/07/2019

$$CF = \frac{\text{Nro. de cambios fallidos}}{\text{Total de cambios cerrados}} * 100$$

FECHA	AREA	Nº CAMBIOS FALLIDOS	TOTAL CAMBIOS CERRADOS	CAMBIOS FALLIDOS
1/07/19	OGETIC	1	7	14.29
2/07/19	OGETIC	1	5	20.00
3/07/19	OGETIC	1	6	16.67
4/07/19	OGETIC	1	8	12.50
5/07/19	OGETIC	2	5	40.00
8/07/19	OGETIC	2	6	33.33
9/07/19	OGETIC	1	7	14.29
10/07/19	OGETIC	3	6	50.00
11/07/19	OGETIC	2	7	28.57
12/07/19	OGETIC	1	6	16.67
15/07/19	OGETIC	2	5	40.00
16/07/19	OGETIC	2	7	28.57
17/07/19	OGETIC	2	6	33.33
18/07/19	OGETIC	2	7	28.57
19/07/19	OGETIC	2	7	28.57

**FICHA DE REGISTRO PARA EL INDICADOR:
“CAMBIOS SIN ERRORES” (POST-TEST)**

INVESTIGADOR	JHONNY PERALTA MORI
EMPRESA	MINISTERIO DE CULTURA
AREA	OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES
FECHA INICIO	01/07/2019
FECHA FIN	19/07/2019

$$CSE = \frac{\text{Nro. de cambios sin errores}}{\text{Total de cambios cerrados}} * 100$$

FECHA	AREA	Nº CAMBIOS SIN ERRORES	TOTAL CAMBIOS CERRADOS	CAMBIOS SIN ERRORES
1/07/19	OGETIC	6	7	85.71
2/07/19	OGETIC	5	5	100.00
3/07/19	OGETIC	5	6	83.33
4/07/19	OGETIC	7	8	87.50
5/07/19	OGETIC	5	5	100.00
8/07/19	OGETIC	5	6	83.33
9/07/19	OGETIC	5	7	71.43
10/07/19	OGETIC	4	6	66.67
11/07/19	OGETIC	5	7	71.43
12/07/19	OGETIC	5	6	83.33
15/07/19	OGETIC	5	5	100.00
16/07/19	OGETIC	6	7	85.71
17/07/19	OGETIC	5	6	83.33
18/07/19	OGETIC	5	7	71.43
19/07/19	OGETIC	6	7	85.71

Validez de los instrumentos



CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura - Lima, 2019.

N°	VARIABLE / DIMENSION	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	Variable independiente: Entrega Continua							
	Dimensión 1: Eficacia	X		X		X		
	Indicador: Frecuencia de Despliegue	X		X		X		
	Dimensión 2: Eficacia	X		X		X		
	Indicador: Cambios Fallidos	X		X		X		
	Dimensión 3: Eficacia	X		X		X		
	Indicador: Cambios sin Errores	X		X		X		

Observaciones (precisar si hay suficiencia): _____

Opinión de aplicabilidad: Aplicable Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador. Dr/ Mg: RODRIGUA BRAVO LIZ MARIHEL DNI: 09217078

Especialidad del validador: METODOLOGA

¹Pertinencia: El ítem corresponde al concepto teórico formulado.
²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.
³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

..... 03 de JUNIO del 2019.....

Firma del Experto Informante.



ESCUELA DE POSTGRADO

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura - Lima, 2019.

Nº	VARIABLE / DIMENSION	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	Variable independiente: Entrega Continua							
	Dimensión 1: Eficacia	X		X		X		
	Indicador: Frecuencia de Despliegue	X		X		X		
	Dimensión 2: Eficacia	X		X		X		
	Indicador: Cambios Fallidos	X		X		X		
	Dimensión 3: Eficacia	X		X		X		
	Indicador: Cambios sin Errores	X		X		X		

Observaciones (precisar si hay suficiencia): _____

Opinión de aplicabilidad: **Aplicable [X]** **Aplicable después de corregir []** **No aplicable []**

Apellidos y nombres del juez validador. Dr/ Mg: Dr. Pedro Martha Lezama Gonzales DNI: 09656793

Especialidad del validador: Iny de sistemas

¹Pertinencia: El ítem corresponde al concepto teórico formulado.
²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.
³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

03 de Junio del 2019



Firma del Experto Informante.



ESCUELA DE POSTGRADO

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura - Lima, 2019.

N°	VARIABLE / DIMENSION	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	Variable independiente: Entrega Continua							
	Dimensión 1: Eficacia	X		X		X		
	Indicador: Frecuencia de Despliegue	X		X		X		
	Dimensión 2: Eficacia	X		X		X		
	Indicador: Cambios Fallidos	X		X		X		
	Dimensión 3: Eficacia	X		X		X		
	Indicador: Cambios sin Errores	X		X		X		

Observaciones (precisar si hay suficiencia): _____

Opinión de aplicabilidad: Aplicable [] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador. Dr/ Mg: PERRYBA ACOSTA MANUEL DNI: 07268838

Especialidad del validador:

¹Pertinencia: El ítem corresponde al concepto teórico formulado.
²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.
³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

03 de Junio del 2019.....


Firma del Experto Informante.

Matriz de datos (Excel y/o spss)

Frecuencia de Despliegues

N	FECHA	Nº DESPLIEGUES REALIZADOS	Nº DESPLIEGUES SOLICITADOS	FRECUENCIA DE DESPLIEGUE	N	FECHA	Nº DESPLIEGUES REALIZADOS	Nº DESPLIEGUES SOLICITADOS	FRECUENCIA DE DESPLIEGUE
1	3/06/2019	3	9	33.33	1	1/07/2019	7	8	87.50
2	4/06/2019	1	7	14.29	2	2/07/2019	5	7	71.43
3	5/06/2019	2	8	25.00	3	3/07/2019	6	7	85.71
4	6/06/2019	3	7	42.86	4	4/07/2019	8	9	88.89
5	7/06/2019	1	8	12.50	5	5/07/2019	5	7	71.43
6	10/06/2019	2	7	28.57	6	8/07/2019	6	8	75.00
7	11/06/2019	3	9	33.33	7	9/07/2019	7	8	87.50
8	12/06/2019	2	8	25.00	8	10/07/2019	6	9	66.67
9	13/06/2019	4	7	57.14	9	11/07/2019	7	7	100.00
10	14/06/2019	3	8	37.50	10	12/07/2019	6	8	75.00
11	17/06/2019	2	8	25.00	11	15/07/2019	5	7	71.43
12	18/06/2019	2	6	33.33	12	16/07/2019	7	8	87.50
13	19/06/2019	2	8	25.00	13	17/07/2019	6	7	85.71
14	20/06/2019	3	7	42.86	14	18/07/2019	7	9	77.78
15	21/06/2019	4	9	44.44	15	19/07/2019	7	8	87.50

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
FD_PRE	,143	15	,200 [*]	,961	15	,713
FD_POST	,218	15	,054	,915	15	,160

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

Prueba T

Estadísticas para una muestra

	N	Media	Desv. Desviación	Desv. Error promedio
FD_PRE	15	32,0100	11,83176	3,05495
FD_POST	15	81,2700	9,25048	2,38846

Prueba para una muestra

Valor de prueba = 0

	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
FD_PRE	10,478	14	,000	32,01000	25,4578	38,5622
FD_POST	34,026	14	,000	81,27000	76,1473	86,3927

Fiabilidad

Escala: ALL VARIABLES

Resumen de procesamiento de casos

Casos		N	%
		Válido	15
Excluido ^a		1	6,3
Total		16	100,0

a. La eliminación por lista se basa en todas las variables del procedimiento.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,812	2

Cambios Fallidos

N	FECHA	Nº CAMBIOS FALLIDOS	TOTAL CAMBIOS CERRADOS	CAMBIOS FALLIDOS	N	FECHA	Nº CAMBIOS FALLIDOS	TOTAL CAMBIOS CERRADOS	CAMBIOS FALLIDOS
1	3/06/2019	2	3	66.67	1	1/07/2019	1	7	14.29
2	4/06/2019	1	1	100.00	2	2/07/2019	1	5	20.00
3	5/06/2019	1	2	50.00	3	3/07/2019	1	6	16.67
4	6/06/2019	2	3	66.67	4	4/07/2019	1	8	12.50
5	7/06/2019	1	1	100.00	5	5/07/2019	2	5	40.00
6	10/06/2019	2	2	100.00	6	8/07/2019	2	6	33.33
7	11/06/2019	2	3	66.67	7	9/07/2019	1	7	14.29
8	12/06/2019	2	2	100.00	8	10/07/2019	3	6	50.00
9	13/06/2019	3	4	75.00	9	11/07/2019	2	7	28.57
10	14/06/2019	2	3	66.67	10	12/07/2019	1	6	16.67
11	17/06/2019	2	2	100.00	11	15/07/2019	2	5	40.00
12	18/06/2019	2	2	100.00	12	16/07/2019	2	7	28.57
13	19/06/2019	2	2	100.00	13	17/07/2019	2	6	33.33
14	20/06/2019	3	3	100.00	14	18/07/2019	2	7	28.57
15	21/06/2019	3	4	75.00	15	19/07/2019	2	7	28.57

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
CF_PRE	,339	15	,000	,766	15	,001
CF_POST	,156	15	,200 [*]	,930	15	,268

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

Prueba T

Estadísticas para una muestra

	N	Media	Desv. Desviación	Desv. Error promedio
CF_PRE	15	84,4453	18,05614	4,66208
CF_POST	15	27,0240	11,18910	2,88901

Prueba para una muestra

Valor de prueba = 0

	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
CF_PRE	18,113	14	,000	84,44533	74,4462	94,4445
CF_POST	9,354	14	,000	27,02400	20,8277	33,2203

Fiabilidad

Escala: ALL VARIABLES

Resumen de procesamiento de casos

		N	%
Casos	Válido	15	93,8
	Excluido ^a	1	6,3
	Total	16	100,0

a. La eliminación por lista se basa en todas las variables del procedimiento.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,803	2

Cambios sin errores

N	FECHA	Nº CAMBIOS SIN ERRORES	TOTAL CAMBIOS CERRADOS	CAMBIOS SIN ERRORES	N	FECHA	Nº CAMBIOS SIN ERRORES	TOTAL CAMBIOS CERRADOS	CAMBIOS SIN ERRORES
1	3/06/2019	1	3	33.33	1	1/07/2019	6	7	85.71
2	4/06/2019	1	1	100.00	2	2/07/2019	5	5	100.00
3	5/06/2019	1	2	50.00	3	3/07/2019	5	6	83.33
4	6/06/2019	2	3	66.67	4	4/07/2019	7	8	87.50
5	7/06/2019	1	1	100.00	5	5/07/2019	5	5	100.00
6	10/06/2019	1	2	50.00	6	8/07/2019	5	6	83.33
7	11/06/2019	1	3	33.33	7	9/07/2019	5	7	71.43
8	12/06/2019	1	2	50.00	8	10/07/2019	4	6	66.67
9	13/06/2019	2	4	50.00	9	11/07/2019	5	7	71.43
10	14/06/2019	1	3	33.33	10	12/07/2019	5	6	83.33
11	17/06/2019	2	2	100.00	11	15/07/2019	5	5	100.00
12	18/06/2019	1	2	50.00	12	16/07/2019	6	7	85.71
13	19/06/2019	1	2	50.00	13	17/07/2019	5	6	83.33
14	20/06/2019	1	3	33.33	14	18/07/2019	5	7	71.43
15	21/06/2019	3	4	75.00	15	19/07/2019	6	7	85.71

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
CSE_PRE	,299	15	,001	,815	15	,006
CSE_POST	,211	15	,072	,893	15	,075

a. Corrección de significación de Lilliefors

Prueba T

Estadísticas para una muestra

	N	Media	Desv. Desviación	Desv. Error promedio
CSE_PRE	15	58,3327	24,60102	6,35196
CSE_POST	15	83,9273	10,57315	2,72998

Prueba para una muestra

Valor de prueba = 0

	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
CSE_PRE	9,183	14	,000	58,33267	44,7091	71,9563
CSE_POST	30,743	14	,000	83,92733	78,0721	89,7826

Fiabilidad

Escala: ALL VARIABLES

Resumen de procesamiento de casos

		N	%
Casos	Válido	15	93,8
	Excluido ^a	1	6,3
	Total	16	100,0

a. La eliminación por lista se basa en todas las variables del procedimiento.

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,734	2

Proyecto / intervención/ Aplicación

En la presente investigación para el DevOps se ha implementado lo siguiente:

La virtualización de recursos, que es uno de los conceptos clave de la computación en la nube y se refiere a la creación de una versión virtual (en lugar de la versión real) de algo, incluyendo, pero no limitado a una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento o recursos de una red informática.

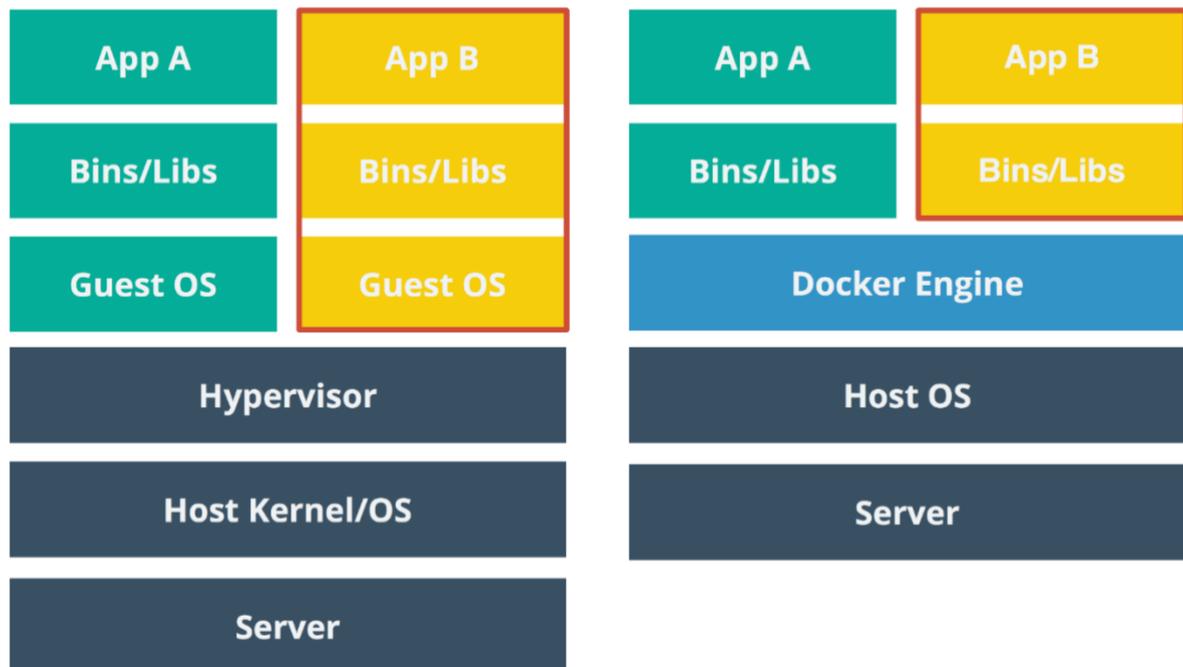
La virtualización hace uso de una capa intermedia de software sobre un sistema para proporcionar abstracción de múltiples recursos virtuales. Actualmente se pueden considerar dos grandes grupos de modelos de virtualización: las máquinas virtuales y los contenedores

Las máquinas virtuales en donde son conocidos como VM y son los recursos virtuales son componentes de software, que podrían ser descritos como contextos de ejecución aislados.

Según Singh & Singh (2016). Entre las técnicas de virtualización que existen en el mercado, una de las más populares es la virtualización basada en hipervisor, que requiere de un Virtual Machine Monitor (VMM) ejecutándose sobre un sistema operativo anfitrión para proporcionar una abstracción completa que permite ejecutar múltiples sistemas operativos en un solo anfitrión real. Ejemplos de virtualización mediante hipervisor incluyen a herramientas como Xen²⁶, KVM²⁷, Virtual Box²⁸ y VM-Ware²⁹.

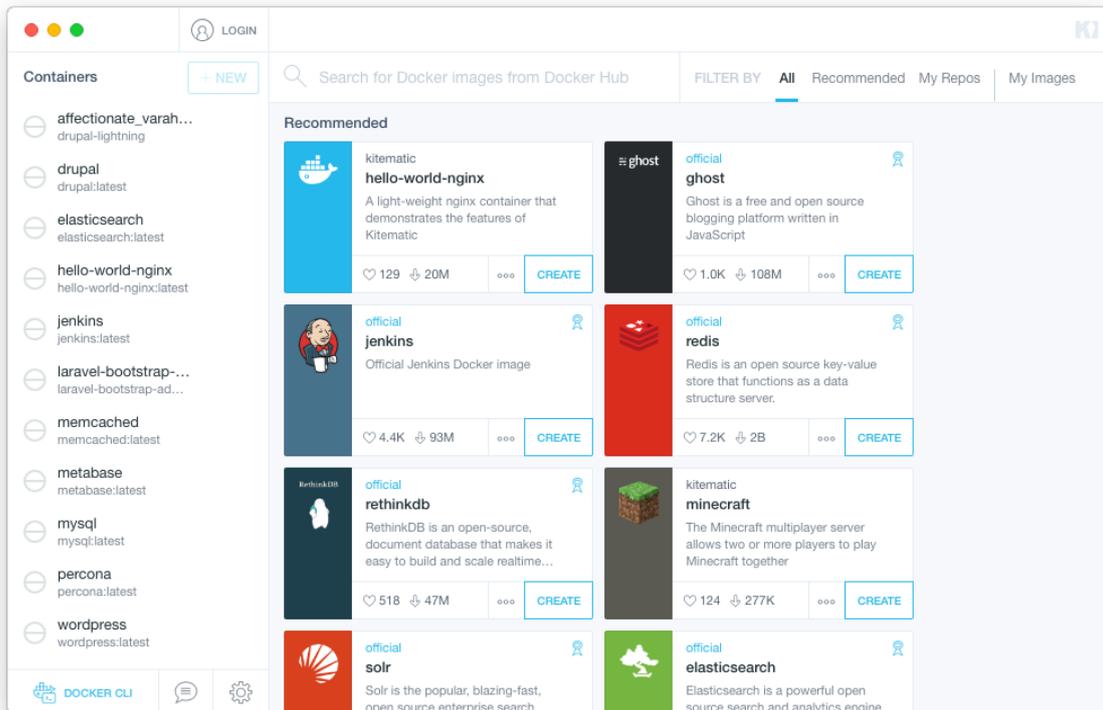
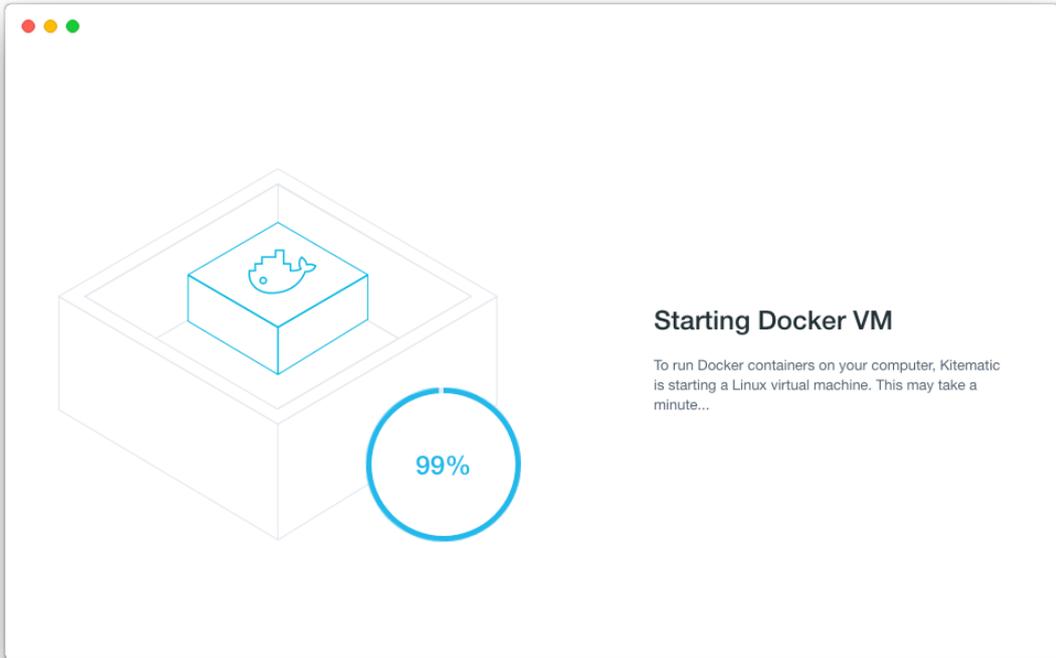
Los contenedores, según Celesti, Mulfari, Fazio, Villari, & Puliafito, (2016), es la técnica de virtualización basada en contenedores (o virtualización de sistema operativo) permite la ejecución de varias máquinas virtuales diferentes sobre el kernel del sistema operativo anfitrión.

Los contenedores comparten un solo kernel del sistema operativo; entonces, la virtualización basada en contenedores se supone que tiene un aislamiento más débil comparado con la virtualización basada en hypervisor según Celesti, Mulfari, Fazio, Villari, & Puliafito (2016). Desde el punto de vista de los usuarios, cada contenedor se ve y ejecuta exactamente como un sistema operativo independiente. Como consecuencia, se pueden desplegar un mayor número de contenedores que máquinas virtuales en un mismo anfitrión físico.



Se muestra la diferencia clave entre las tecnologías de virtualización. Mientras que la virtualización basada en hypervisor proporciona abstracción para sistemas operativos invitados completos (uno por VM), la virtualización basada en contenedores funciona a nivel del sistema operativo proporcionando abstracciones directamente para los procesos de invitado según Li, Kihl, Lu, & Andersson (2017)

En la presente investigación se implementó el contenedor Docker, porque toda vez que en esencia, las soluciones de hypervisor funcionan como abstracción a nivel de hardware mientras que la virtualización basada en contenedores opera a nivel de llamada del sistema proporcionando mejores beneficios.



Así mismo se implementa el sistema de versión de controles. Según Just, Herzig, Czerwonka, & Murphy (2016). Los sistemas de control de versiones modernos están implementados con una arquitectura distribuida, facilitando la adopción de metodologías ágiles en los equipos de desarrollo a través de procesos de integración más rápidos, y mejorando el aislamiento entre los desarrolladores para garantizar la independencia de los cambios.



GitLab

En la presente investigación se implemento el sistema de control de versiones GitLab, el cual cuenta con la fuente de los proyectos web, de esta manera los cambios realizados se puede verificar que línea se a cambiado, actualizado, eliminado, agregado.



GitLab Enterprise Edition

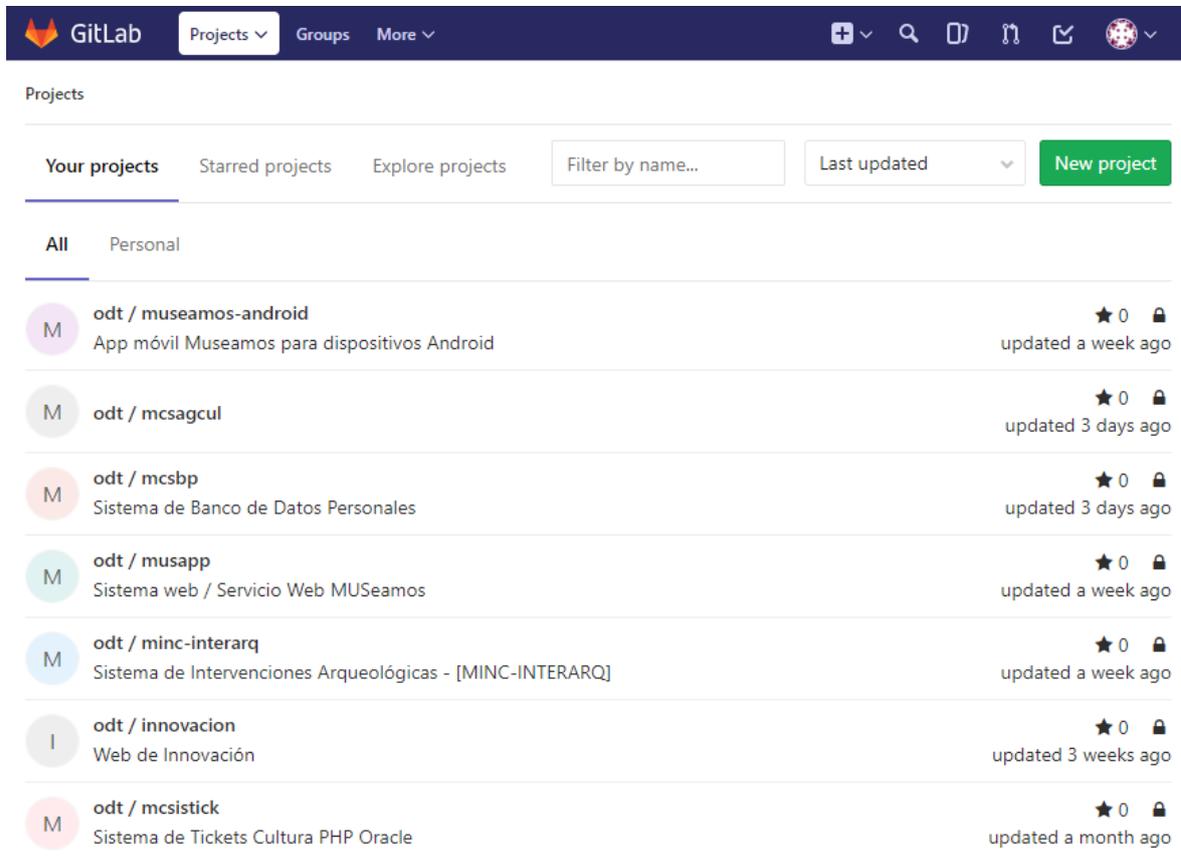
Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in	Register
Username or email	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	Forgot your password?
<input type="button" value="Sign in"/>	

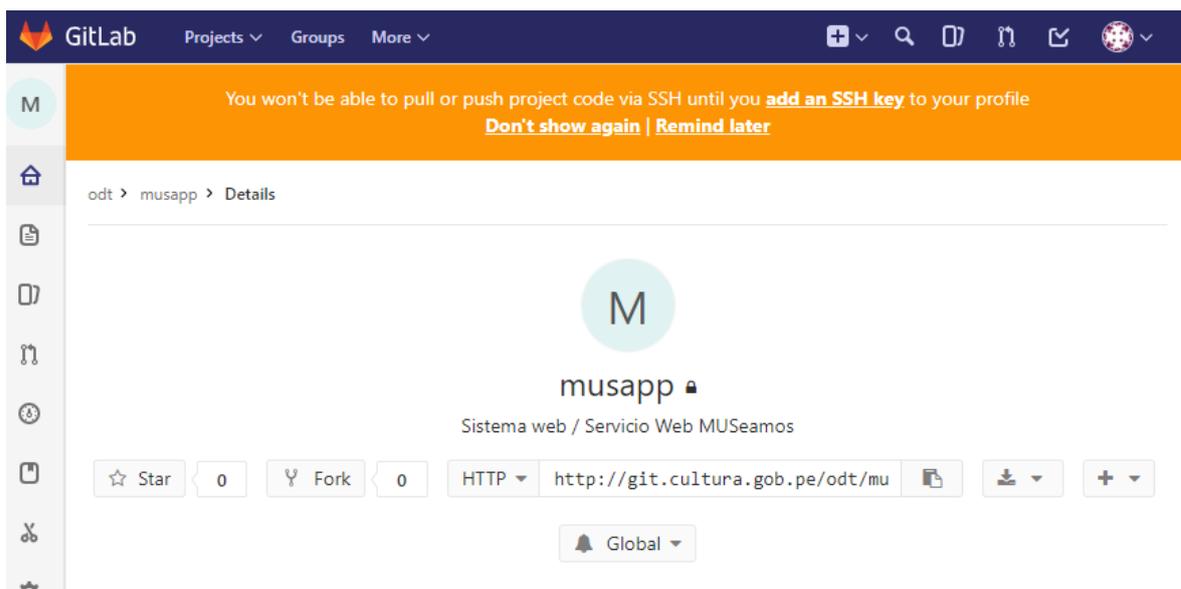
Didn't receive a confirmation email? [Request a new one.](#)

Se usa el sistema de control de versiones GitLab en donde ofrece mejoras en la entrega continua al tener todo los registros de los codigos y poder realizar los pases a producción de una manera transparente.



The screenshot shows the GitLab interface for the 'Projects' section. The top navigation bar includes the GitLab logo, 'Projects', 'Groups', and 'More' menus, along with utility icons for adding, searching, cloning, forking, and notifications. Below the navigation, there are tabs for 'Your projects', 'Starred projects', and 'Explore projects', a search filter, a 'Last updated' dropdown, and a 'New project' button. The main content area displays a list of projects under the 'All' tab, with 'Personal' also visible. Each project entry includes a profile icon, the project name and description, a star count, a lock icon, and the last update time.

Profile	Project Name	Description	Stars	Lock	Last Updated
M	odt / museamos-android	App móvil Museamos para dispositivos Android	0	Yes	updated a week ago
M	odt / mcsagcul		0	Yes	updated 3 days ago
M	odt / mcsbp	Sistema de Banco de Datos Personales	0	Yes	updated 3 days ago
M	odt / musapp	Sistema web / Servicio Web MUSeamos	0	Yes	updated a week ago
M	odt / minc-interarq	Sistema de Intervenciones Arqueológicas - [MINC-INTERARQ]	0	Yes	updated a week ago
I	odt / innovacion	Web de Innovación	0	Yes	updated 3 weeks ago
M	odt / mcsistick	Sistema de Tickets Cultura PHP Oracle	0	Yes	updated a month ago



The screenshot shows the details page for the 'musapp' project. At the top, there is an orange notification banner stating: 'You won't be able to pull or push project code via SSH until you **add an SSH key** to your profile. Don't show again | Remind later'. Below the notification, the breadcrumb 'odt > musapp > Details' is visible. The main content area features a large profile icon with the letter 'M', the project name 'musapp' with a lock icon, and the description 'Sistema web / Servicio Web MUSeamos'. Below this, there are buttons for 'Star' (0), 'Fork' (0), and the HTTP clone URL 'http://git.cultura.gob.pe/odt/mu'. There are also icons for cloning, downloading, and adding a new item. At the bottom, there is a 'Global' notification bell icon.

Adicional se esta usando a Composer, que es un administrador de paquetes a nivel de aplicación para el lenguaje de programación PHP que proporciona un formato estándar para administrar las dependencias del software PHP y las bibliotecas requeridas.

```
Composer version 1.8.3 2019-01-30 08:31:33

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -vv|vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
```

```
Available commands:
  about                    Shows the short information about Composer.
  archive                  Creates an archive of this composer package.
  browse                   Opens the package's repository URL or homepage in your browser.
  check-platform-reqs     Check that platform requirements are satisfied.
  clear-cache              Clears composer's internal package cache.
  clearcache               Clears composer's internal package cache.
  config                   Sets config options.
  create-project           Creates new project from a package into given directory.
  depends                  Shows which packages cause the given package to be installed.
  diagnose                 Diagnoses the system to identify common errors.
  dump-autoload            Dumps the autoloader.
  dumpautoload             Dumps the autoloader.
  exec                     Executes a vendored binary/script.
  global                   Allows running commands in the global composer dir ($COMPOSER_HOME).
  help                     Displays help for a command
  home                     Opens the package's repository URL or homepage in your browser.
  i                         Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json.
  info                     Shows information about packages.
  init                     Creates a basic composer.json file in current directory.
  install                  Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json.
  licenses                 Shows information about licenses of dependencies.
  list                     Lists commands
  outdated                Shows a list of installed packages that have updates available, including their latest version.
  prohibits                Shows which packages prevent the given package from being installed.
  remove                   Removes a package from the require or require-dev.
  require                  Adds required packages to your composer.json and installs them.
  run-script               Runs the scripts defined in composer.json.
  search                   Searches for packages.
  self-update              Updates composer.phar to the latest version.
  selfupdate              Updates composer.phar to the latest version.
  show                     Shows information about packages.
  status                   Shows a list of locally modified packages, for packages installed from source.
  suggests                 Shows package suggestions.
  u                         Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock file.
  update                  Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock file.
  upgrade                  Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock file.
  validate                Validates a composer.json and composer.lock.
  why                      Shows which packages cause the given package to be installed.
  why-not                 Shows which packages prevent the given package from being installed.
```

También se usa Drush que es una aplicación basada en shell de software de computadora utilizada para controlar, manipular y administrar sitios web de Drupal. En la superficie, drush es una herramienta para actualizar los módulos del sitio, sin embargo, Drush tiene una lista más completa de características

```
Drush Commandline Tool 9.5.2

Run `drush help [command]` to view command-specific help. Run `drush topic` to read even more documentation.

Available commands:
_global:
  help          Display usage details for a command.
  list          List available commands.
  browse        Display a link to a given path or open link in a browser.
  version       Show Drush version.
  runserver (rs, serve)  Runs PHP's built-in http server for development.
  updatedb (updb)  Apply any database updates required (as with running update.php).
  generate (gen)  Generate boilerplate code for modules/plugins/services etc.
  config:pull (cpull)  Export and transfer config from one environment to another.
  drupal:directory (dd)  Return the filesystem path for modules/themes and other key folders.
  user:login (uli)  Display a one time login link for user ID 1, or another user.
  entity:updates (entup)  Apply pending entity schema updates.
  updatedb:status (updbst)  List any pending database updates.
  pm:security (sec)  Check Drupal Composer packages for pending security updates.
cache:
  cache:get (cg)  Fetch a cached object and display it.
  cache:clear (cc)  Clear a specific cache, or all Drupal caches.
  cache:set (cs)  Cache an object expressed in JSON or var_export() format.
  cache:rebuild (cr, rebuild)  Rebuild a Drupal 8 site.
core:
  core:execute (exec, execute)  Execute a shell command. Usually used with a site alias.
  core:edit (conf, config)  Edit drushrc, site alias, and Drupal settings.php files.
  core:init (init)  Enrich the bash startup file with bash aliases and a smart command prompt.
  core:rsync (rsync)  Rsync Drupal code or files to/from another server using ssh.
  core:status (status, st)  An overview of the environment - Drush and Drupal.
  core:topic (topic)  Read detailed documentation on a given topic.
php:
  php:eval (eval, ev)  Evaluate arbitrary php code after bootstrapping Drupal (if available).
  php:script (scr)  Run php a script after a full Drupal bootstrap.
site:
  site:set (use)  Set a site alias that will persist for the current session.
  site:alias (sa)  Show site alias details, or a list of available site aliases.
  site:alias-convert (sa-convert, sac)  Convert legacy site alias files to the new yml format.
  site:install (si, sin)  Install Drupal along with modules/themes/configuration/profile.
  site:ssh (ssh)  Connect to a Drupal site's server via SSH.

sql:
  sql:connect  A string for connecting to the DB.
  sql:create  Create a database.
  sql:drop  Drop all tables in a given database.
  sql:cli (sqlc)  Open a SQL command-line interface using Drupal's credentials.
  sql:query (sqlq)  Execute a query against a database.
  sql:dump  Exports the Drupal DB as SQL using mysqldump or equivalent.
  sql:sync  Copy DB data from a source site to a target site. Transfers data via rsync.
```

Para los proyectos web se usa Drupal, que si bien es un sistema de gestión de contenidos o CMS libre, también calza como un Framework toda vez que es modular, multipropósito y muy configurable que permite publicar artículos, imágenes, archivos y que también ofrece la posibilidad de otros servicios añadidos como foros, encuestas, votaciones, blogs, administración de usuarios y permisos, así mismo conectarse y operar con sistemas de información y aplicaciones móviles mediante servicios.



Ejemplo de ejecución de códigos:

Despliegue de Aplicaciones: Actualización

Paso 1	Ingresar a la terminal del servidor de producción
Paso 2	Ubicarse en el directorio raíz del portal web
Paso 3	<p>Ingresar los siguientes comandos:</p> <pre> mv vendor vendor_bk mv composer.lock composerlock_bk composer clear-cache composer install composer update drupal/core --with-dependencies drush updatedb drush cr </pre>

Autorización



UNIVERSIDAD CÉSAR VALLEJO

Escuela de Posgrado

"Año de la lucha contra la corrupción y la impunidad"

Lima, 16 de julio de 2019

Carta P. 0306-2019-EPG-UCV-LN

ING. JOAN MIGUEL PALACIOS RAMIREZ
DIRECTOR DE LA OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y
COMUNICACIONES
Ministerio de Cultura

De mi mayor consideración:

Es grato dirigirme a usted, para presentar a JHONNY PERALTA MORI identificado con DNI N.° 70143675 y código de matrícula N.° 6700248225; estudiante del Programa de MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN quien se encuentra desarrollando el Trabajo de Investigación (Tesis):

**DEVOPS EN LA ENTREGA CONTINUA DE LA OFICINA GENERAL DE ESTADÍSTICA Y
TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES DEL MINISTERIO DE
CULTURA, LIMA 2019**

En ese sentido, solicito a su digna persona otorgar el permiso y brindar las facilidades a nuestro estudiante, a fin de que pueda desarrollar su trabajo de investigación en la institución que usted representa. Los resultados de la presente serán alcanzados a su despacho, luego de finalizar la misma.

Con este motivo, le saluda atentamente,



Dr. Carlos Ventura Orbegoso
Jefe de la Escuela de Posgrado
Universidad César Vallejo - Campus Lima Norte

RCOA

Ministerio de Cultura
Oficina de Desarrollo Tecnológico

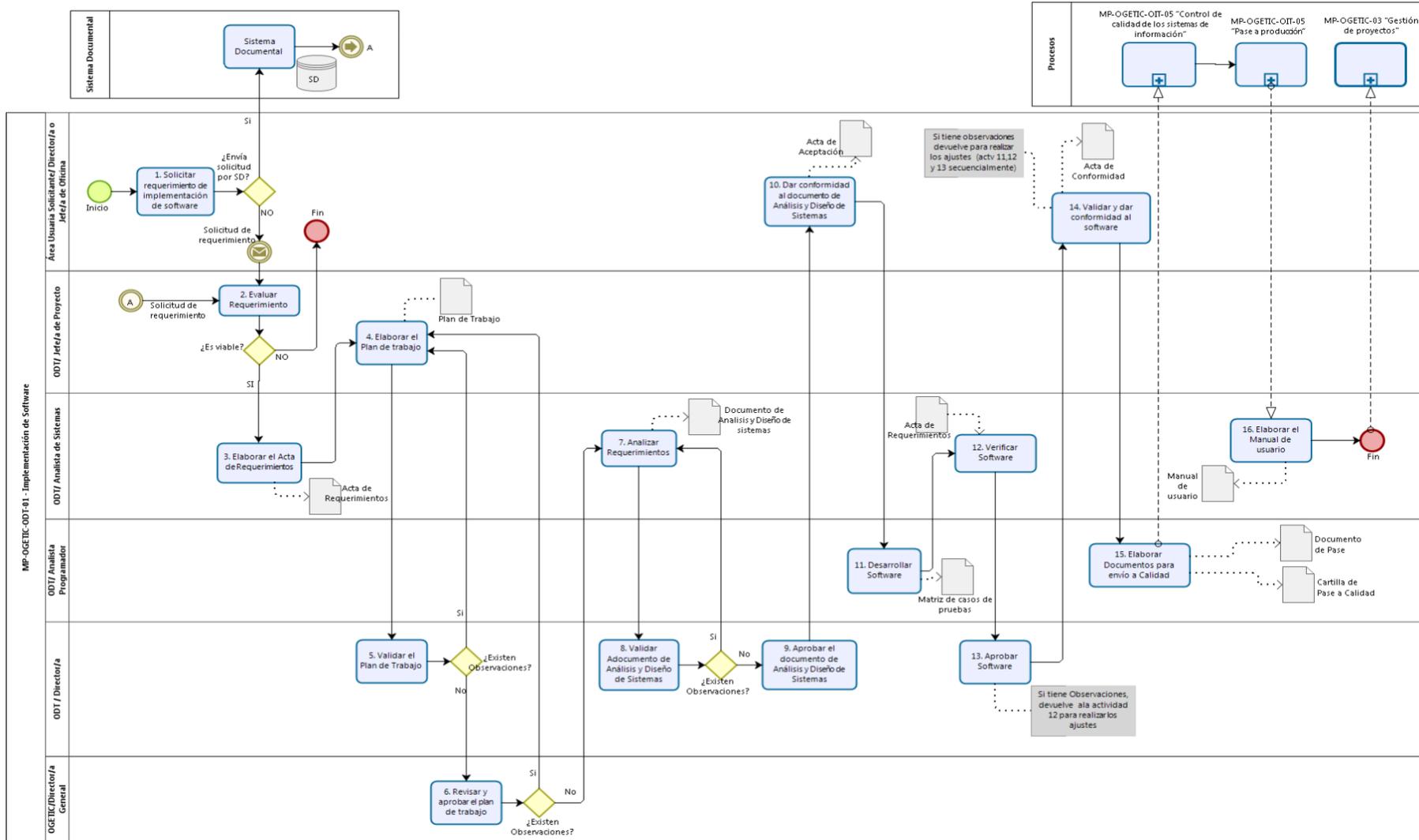
Willy David Yucra Limahuay
Director

Somos la universidad de los
que quieren salir adelante.



ucv.edu.pe

Proceso



Acta de Aprobación de originalidad de Tesis

Yo, Liz Maribel Robladillo Bravo, docente de la Escuela de Posgrado de la Universidad César Vallejo filial Lima Norte, revisor de la tesis titulada "DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, Lima 2019" del estudiante **Peralta Mori Jhonny** constato que la investigación tiene un índice de similitud de 22% verificable en el reporte de originalidad del programa Turnitin.

La suscrita analizó dicho reporte y concluyó que cada una de las coincidencias detectadas no constituye 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.

Lima, 23 de setiembre de 2019



Liz Maribel Robladillo Bravo

DNI:09217078

Preparando la



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

DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, Lima 2019

TESIS PARA OBTENER EL GRADO ACADÉMICO DE:
Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información

AUTOR:
Bch. Jhonny Peralta Mori
(ORCID: 0000-0002-7032-5041)

ASESOR:
Dr. Liz Robladillo Bravo
(ORCID: 0000-0000-0000-0000)

Resumen de coincidencias

22 %

Se están viendo fuentes estándar

Ver fuentes en inglés (Beta)

Coincidencias

- 1 Entregado a Universida... 9 % >
Trabajo del estudiante
- 2 repositorio.ucv.edu.pe 5 % >
Fuente de Internet
- 3 dspace.ucuenca.edu.ec 3 % >
Fuente de Internet
- 4 Entregado a Universida... 1 % >
Trabajo del estudiante
- 5 www.javiercanon.com <1 % >
Fuente de Internet
- 6 myslide.es <1 % >
Fuente de Internet



FORMULARIO DE AUTORIZACIÓN PARA LA PUBLICACIÓN ELECTRÓNICA DE LAS TESIS

1. DATOS PERSONALES

Apellidos y Nombres: (solo los datos del que autoriza)

PERALTA MORI JHONNY

D.N.I. : 70143675

Domicilio : J.R. JOSE PEREYRA 546 LOS OLIVOS

Teléfono : Fijo : Móvil : 985113825

E-mail : JHONNY@MORISYS.COM

2. IDENTIFICACIÓN DE LA TESIS

Modalidad:

Tesis de Pregrado

Facultad :

Escuela :

Carrera :

Título :

Tesis de Posgrado

Maestría

Doctorado

Grado : MAESTRO EN INGENIERIA DE SISTEMAS

Mención: TECNOLOGÍAS DE LA INFORMACION

3. DATOS DE LA TESIS

Autor (es) Apellidos y Nombres:

PERALTA MORI JHONNY

Título de la tesis:

DEVOPS EN LA ENTREGA CONTINUA DE LA OFICINA
GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN
Y COMUNICACIONES DEL MINISTERIO DE CULTURA, UAA 2019

Año de publicación : 2019

4. AUTORIZACIÓN DE PUBLICACIÓN DE LA TESIS EN VERSIÓN ELECTRÓNICA:

A través del presente documento, autorizo a la Biblioteca UCV-Lima Norte, a publicar en texto completo mi tesis.

Firma :

Fecha : 27/11/2019



UNIVERSIDAD CÉSAR VALLEJO

AUTORIZACIÓN DE LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN

CONSTE POR EL PRESENTE EL VISTO BUENO QUE OTORGA EL ENCARGADO DE INVESTIGACIÓN DE

ESCUELA DE POSGRADO

A LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN QUE PRESENTA:

JHONNY PERALTA MORI

INFORME TITULADO:

DEVOPS EN LA ENTREGA CONTINUA DE LA OFICINA GENERAL DE ESTADÍSTICA Y TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIONES DEL MINISTERIO DE CULTURA, LIMA 2019

PARA OBTENER EL TÍTULO O GRADO DE:

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

SUSTENTADO EN FECHA: 16 DE AGOSTO DE 2019

NOTA O MENCIÓN: APROBADO POR MAYORÍA



FIRMA DEL ENCARGADO DE INVESTIGACIÓN