

# DevOps como agente de cambio para las organizaciones que buscan una cultura ágil

Jose Luis Mejías Artavia  
Facultad de TI

ULACIT  
San José, Costa Rica  
[jmejiasa923@ulacit.ed.cr](mailto:jmejiasa923@ulacit.ed.cr)

Johel Elizondo Gutiérrez  
Facultad de TI

ULACIT  
San José, Costa Rica  
[jelizondog531@ulacit.ed.cr](mailto:jelizondog531@ulacit.ed.cr)

Steven Guevara Carrillo  
Facultad de TI

ULACIT  
San José, Costa Rica  
[sguevarac855@ulacit.ed.cr](mailto:sguevarac855@ulacit.ed.cr)

Julio Córdoba Retana  
Facultad de TI

ULACIT  
San José, Costa Rica  
[jcordobar022@ulacit.ed.cr](mailto:jcordobar022@ulacit.ed.cr)

**Resumen.** Las exigencias actuales en el desarrollo de *software* crean la necesidad de implementar un conjunto de buenas prácticas con metodologías ágiles, que ayuden a hacer este proceso aún más eficiente. DevOps está fuertemente asociado con enfoques ágiles y continuos de desarrollo de *software* siguiendo buenas prácticas de aceptación e integración, da las prioridades correspondientes según las estrategias implicadas con metodologías ágiles. Por lo cual, se ha analizado lo que implica DevOps y cómo esto conlleva a una mejoría en los tiempos de desarrollo e implementación.

**Abstract.** The demanding needs that currently exist in software development create the need to implement a set of good practices with agile methodologies that help make this process even more efficient. Where devOps is strongly associated with agile and continuous approaches to software development following good acceptance and integration practices, giving the corresponding priorities according to the strategies involved with agile methodologies. They have analyzed what DevOps implies and how this leads to an improvement in development and implementation times.

**Palabras claves:** DevOps, Scrum, *software*, implementación, integración.

**Keywords:** DevOps, Scrum, *software*, implementation, integration.

## I. INTRODUCCIÓN

En los últimos años, la industria del *software* ha evolucionado constantemente, mejorando la calidad de los procesos y de los productos entregados; los cuales dependen de los requerimientos y las necesidades de los clientes. Con el fin de obtener una respuesta a estos cambios, han surgido diferentes herramientas y marcos de trabajo que buscan agilizar cada uno de los procesos, generando un mayor valor agregado, calidad de *software* y reduciendo los tiempos de espera para percibir el retorno de la inversión.

Dichas herramientas y marcos de trabajo han aumentado la productividad de los equipos de desarrollo notablemente, permitiéndole a las organizaciones crear soluciones o mejoras a los sistemas en el menor tiempo posible y con una capacidad de reacción más temprana ante cualquier situación que se presente. No obstante, esta agilidad que han ganado las organizaciones con la implementación de metodologías ágiles como Scrum ha generado cuellos de botella a la hora de implementar estos cambios, debido a la alta productividad de los equipos de desarrollo, la falta de comunicación entre las fábricas de *software*, la ausencia de procesos y herramientas que faciliten la integración de estas soluciones en los ambientes de producción; provoca un aumento en los tiempos de espera en la puesta de producción de los desarrollos, frenando el avance de la organización y haciendo menos eficiente y rentable el valor agregado que aporta Scrum a las organizaciones.

La combinación del marco de trabajo Scrum y las buenas prácticas de DevOps ha demostrado tener grandes beneficios para las organizaciones, permitiéndoles mantenerse al paso de los cambios constantes en la industria. DevOps es una forma de abordar la cultura de transformación digital y diseño de plataformas empresariales con mayor valor agregado y capacidad de respuesta, mediante la implementación de procesos ágiles y servicio de alta calidad.

DevOps es la combinación de las palabras *development* que significa desarrollo y *operations* que significa operaciones; incluye en sus ámbitos de implementación seguridad, maneras de trabajar en sinergia con otros equipos, análisis de datos, entre otros [1].

Hoy en día, DevOps es de los términos más mencionados en la industria de tecnologías de la información, el cual está muy ligado con estrategias de transformación digital, metodologías de desarrollo ágil y entrega continua de producto. Promueve una comunicación más fluida, colaboración, integración y transparencia entre los equipos de desarrollo de *software* y los encargados de operaciones de tecnologías de información (TI). Dicha relación se extiende en cada una de las fases de su ciclo de vida, desde el momento en que se diseña el *software*, se desarrolla, compila, prueba y publica en ambientes productivos [2].

DevOps se encarga de la integración y entrega continua del *software*, donde se plantea, mediante el uso de un conjunto de buenas prácticas, liberaciones en ambientes de producción de manera rápida y confiable, reduciendo los riesgos que se generan.

La implementación de DevOps mejora el rendimiento de los equipos de trabajo; la creación de productos de calidad en menor tiempo aumenta la satisfacción de los clientes, mejora la colaboración y la productividad. Algunos de los beneficios que aporta son: reducción del tiempo de comercialización, adaptación al mercado y a la competencia, mantenimiento de la estabilidad y la confiabilidad del sistema, así como mejora el tiempo medio de recuperación [3].

DevOps aporta una serie de ventajas técnicas para los equipos y las organizaciones que adoptan su uso; se observan mejoras en el proceso de desarrollo, se acortan los tiempos de publicación, se mejora la respuesta ante los incidentes en producción y la calidad de las aplicaciones mejora considerablemente permitiendo así la posibilidad de crear sistemas escalables y seguros. Además, gracias a que uno de sus pilares es la comunicación, propicia entornos colaborativos, donde los equipos trabajan con total sinergia en busca de un objetivo en común [4].

Al implementar DevOps en la organización, se llega a tener una serie de cambios que mejoran diversos aspectos; aun así, se debe mejorar la parte de la interacción de los equipos con los clientes; ante esta problemática, se debe considerar la implementación de Scrum. Según la página oficial de Scrum, se define como un marco que se enfoca en proporcionar soporte a las personas, equipos y organizaciones, a dar valor a las soluciones para diversos problemas que son complejos. Se puede adaptar a una variedad de procesos, técnicas y métodos. Tomando esto en cuenta, se plantea cómo las distintas prácticas Scrum pueden llegar a integrarse con DevOps para así aprovechar de mejor manera los beneficios que ofrecen [5].

La integración entre Scrum y DevOps se basa en los desarrollos incrementales, ya que, primero, se analizan los requerimientos, se ejecuta y se da la solución correspondiente al problema planteado, generando desarrollos más robustos y estables; todo esto mediante un desempeño óptimo de la exploración continua (CE), la cual se centra en cómo el equipo debe interactuar, donde cada aporte debe ser honesto y dar un crecimiento a cada integrante. Se enfoca en que cada miembro del equipo debe ver su pasado, aprender de los errores y tener un mejor desempeño en la siguiente iteración.

La integración continua (CI) es la forma como cada componente o funcionalidad creada por un desarrollador se integra al proyecto. El despliegue continuo (CD) fomenta el despliegue frecuente de cambios de bajo riesgo en ambientes de producción; comprende despliegues, alternancia de funciones, automatización de implementación, despliegues automáticos y se necesita control de versiones.

La liberación bajo demanda del *software* (*Release on demand*) hace referencia a las integraciones bajo demanda y, dependiendo de la función, el usuario podrá acceder de manera inmediata a la nueva característica, sin percibir una afectación en el servicio o sistema, haciendo transparente la implementación de nuevas características en el *software* [6].

Para lograr una adecuada integración de DevOps, debe generarse una cultura colaborativa, emplear Scrum u otra metodología ágil; establecer métricas, adoptar las fases de integración y despliegue continuos, así como emplear las herramientas adecuadas [7]. Otro aspecto fundamental son las herramientas y cómo se emplean en cada una de las fases, lo cual permite obtener un óptimo desempeño.

DevOps se compone de diferentes fases, las cuales interactúan en sinergia con los ciclos de vida del *software*, apoyándose en el uso de herramientas que hacen mucho más eficiente cada uno de sus procesos. Las fases que componen el ciclo de DevOps son: planificar, en esta fase se define y diseña el software por construir, se identifica la problemática del cliente y cómo se abordará; además se define la forma en la que se gestionará el proyecto.

La segunda fase es el desarrollo, en la cual se lleva a cabo la codificación, revisión, integración y despliegue del *software* de manera ágil y colaborativa de forma iterativa. En la fase de construcción, se realiza la liberación del código en los diferentes ambientes, generando paquetes para mantener un estándar, haciendo uso de herramientas. La fase de pruebas consiste en la ejecución de los casos de pruebas, con el fin de encontrar alguna falla que pueda afectar el rendimiento y la calidad del *software*.

Por su parte, en la fase de liberación, se da la integración de las nuevas funcionalidades con las ya existentes y en el despliegue, es cuando el *software* es lanzado a los ambientes productivos continuamente, de forma fiable y coherente. Para finalizar, la fase de operación se basa en el uso del *software* liberado, su gestión y atención de incidentes; además, el monitoreo es una fase permanente, en la que se supervisa, da mantenimiento, se registran los errores y se controla el estado de salud del *software* [8].

## II. MARCO TEÓRICO

DevOps es un término relativamente nuevo, que aparece en la industria en el año 2009; hoy en día, muchas de las grandes y exitosas empresas como Etsy, Facebook, Amazon y Netflix son líderes en su implementación. La finalidad de este paradigma es la reducción de los tiempos de espera que pasan desde que se sube un cambio al repositorio de control de versiones, hasta el momento en que esto es llevado a producción por la entidad; lo cual aumenta la cantidad de mejoras que se llevan a producción en un mes, generando sistemas más robustos y confiables, lo que se ve reflejado en el retorno de las ganancias para la organización [9].

DevOps ha significado para las organizaciones un aumento de ingresos y clientes, lo cual se ha traducido en un incremento del 20 % de los negocios; esto demuestra que realmente existe una oportunidad de mejora y que esta ha sido abordada por el paradigma. Por otro lado, es una práctica que puede ser adoptada por empresas de cualquier tamaño, donde, mediante los despliegues continuos, la reducción de complejidad de la administración y resolución de problemas, se pueda obtener equipos de trabajo más productivos, mayor compromiso de los empleados y oportunidades de crecimiento profesional para estos [9].

En el informe de estado de DevOps del 2021, indica que las organizaciones más evolucionadas se benefician de la habilitación de la transformación digital y que solamente el 2 % de las organizaciones muestran resistencia a DevOps; lo cual indica que, cada vez más, las organizaciones reconocen los beneficios de DevOps y optan por su adopción [10].

Como ya se ha mencionado, DevOps también es comunicación, lo que implica que se abra un canal de comunicación entre los departamentos de una entidad de manera más fluida, monitoreando el flujo de trabajo en ambas direcciones, generando una constante retroalimentación entre los departamentos de desarrollo y operaciones. Además, como parte de los beneficios que ofrece el marco de trabajo, se mejora la calidad, seguridad gracias al versionado y análisis del código fuente; se mejora notablemente la escalabilidad, debido a que aporta flexibilidad para añadir funcionalidades, pudiendo implementar versiones de forma paralela sin arriesgar la producción del sistema [11].

Por otro lado, aparte de maximizar la rentabilidad y reducir los costes de operación, DevOps permite a las organizaciones mantener una mejora continua a la experiencia del usuario, mediante el monitoreo de los sistemas y la implementación de la mejora continua en su ciclo de vida infinito; lo cual permite aprovechar de mejor manera las habilidades de los colaboradores, esto mediante la innovación, implementación y adaptación de nuevas herramientas y procesos que permiten aumentar la calidad de los productos [11].

Al trabajar la cultura DevOps en la organización y si esta dispone el marco de trabajo Scrum, deben acoplarse ambas formas para tener una implementación en el desarrollo y despliegue de las aplicaciones que cumpla con los mayores estándares de calidad. Cada metodología dispone de una serie de buenas prácticas. Cabe mencionar que cada una de estas prácticas dispone de un objetivo que debe cumplirse y entre las prácticas DevOps, se encuentran: exploración continua (CE), integración continua (CI), despliegue continuo (CD) y liberación bajo demanda (Release on Demand).

La exploración continua se centra en cómo el equipo debe trabajar de manera conjunta, donde se generan espacios seguros para aumentar el conocimiento basado en experiencias y en dar sugerencias a los demás miembros del equipo. En el equipo, los roles que llegan a tener un cambio a nivel de tareas

son el dueño del producto (*Product Owner*), gerente, arquitecto, ingeniero de producción y encargado de lanzamientos [12]; el rol de programador o desarrollador no sufre un cambio, dado que su tarea principal es escribir código.

Otro aspecto esencial para el equipo es el balance que debe existir entre las habilidades técnicas y las blandas, debido a que se necesita que todos los involucrados sean capaces de transmitir sus ideas al equipo y que cada miembro pueda trabajar de manera independiente, si está a cargo del proyecto. Por lo tanto, las habilidades blandas requeridas son liderazgo, toma de decisiones, colaboración y comunicación; mientras que las habilidades técnicas se relacionan al proyecto en sí, como programar en un determinado lenguaje. Sin embargo, esto puede cambiar, dado que, al implementar la práctica de exploración continua, ocurre un cambio en las habilidades técnicas, donde cada miembro dispone de un rol y competencias específicas, pero deberá ser capaz de adaptarse a varias áreas.

Integración continua tiene como objetivo cómo el programador agrega las nuevas líneas de código al código existente; esto puede ser un módulo, una función, un objeto o la modificación de algún elemento existente. Para verificar y validar la consistencia del código, se emplea un servidor de compilación, lo que da como resultado que, cada vez que se detecta un cambio en la base del código, se inicia una compilación que prueba la calidad del código recién enviado [13]; tomando en cuenta que cada desarrollador puede disponer de una versión diferente del código y debe validarse los cambios en todas las versiones.

Debe considerar que cada versión del código de cada desarrollador no debe presentar un distanciamiento tan largo basado en los cambios integrados. Al presentarse este escenario, sucede que, a la hora de validar los últimos cambios, suceden diversos errores y no dejará que todas las versiones se actualicen de manera conjunta.

Es importante buscar una herramienta que permita llevar un control de las versiones, ver las distintas ramas y, por medio de una acción, se actualicen todas las ramas. Debe velar por todos los cambios realizados por muchas personas a lo largo de los años. Estos incluyen la creación y la eliminación de los archivos, así como los cambios de sus contenidos [14]; tomando el proceso de auditar los cambios, se volverá un proceso transparente y sencillo; se identificará el efectuado y, en caso de que no cumpla con los criterios, volverá un estado anterior, ya que esto puede significar un ahorro importante en los costos del código.

Despliegue continuo se centra en cómo se integra el código en ambientes de producción, cómo se controla el despliegue para disminuir la mayor cantidad de errores, tener un control de versiones y este proceso debe ser automatizado. Un aspecto fundamental de este proceso es la duración que transcurre desde el ambiente de pruebas al de producción. Dado esto, es más fácil recordar lo que hizo ayer, que pudo haber causado el

error; que recordar lo que hizo hace dos meses, que pudo haberlo causado [15]. Esto indica que, al hacer el proceso, si aparece un error en el ambiente de producción, será sencillo para el programador encontrar una solución, dado que tendrá presente la funcionalidad y hallará rápidamente una solución en el ambiente de producción. Ambos ambientes deben disponer de características similares para que la integración no sea tan difícil.

Sin embargo, esto casi no sucede, dado que muchos administradores o líderes de proyectos temen hacer este proceso e incluso realizarlo de manera automatizada, ya que al generarlo de forma manual, es más propenso a errores. El flujo que pasa el *software* en esta práctica incluye un entorno de prueba, un entorno de integración y, a veces, un entorno de preproducción. En última instancia, si todo sale según lo planeado (sin ninguna intervención manual), esta nueva compilación se implementará en producción [16].

La práctica integración y despliegue continuo es un proceso que debe ejecutarse de forma automática, donde incluye pruebas de código, se integra a las diferentes versiones, se genera una nueva versión, la cual se lleva al ambiente de producción y dará una nueva versión del programa.

Liberación bajo demanda es cuando se logró generar la nueva versión del código en el ambiente de producción; debe indicar qué funcionalidades estarán disponibles y en qué momento para los usuarios finales, incluso, puede determinar también los usuarios que dispondrán el acceso. Ante esto, se sugiere que puede decidir habilitar la nueva función solo para un pequeño porcentaje de sus usuarios, con el fin de ver si la nueva función se ejecuta correctamente [16]. Con esta recomendación, se puede mejorar muchos aspectos de la experiencia de usuario como la funcionalidad y cuando se disponga para todo el mercado, irá acorde a las necesidades de un grupo representativo de usuarios.

Al disponer de una claridad con las distintas prácticas DevOps que se deben considerar, debido a que debe implementarse con la metodología ágil Scrum, es necesario considerar los diversos eventos que Scrum posee. Dichos eventos son planificación del *sprint*, Scrum diario, reunión de revisión de *sprint*, reunión retrospectiva de *sprint* y reuniones de refinamiento de reserva (*Backlog*) [17]. Al tomar en cuenta estos eventos, se puede identificar cuál práctica DevOps se asocia a una etapa de Scrum. Así mismo, se debe considerar que, para empezar un nuevo evento en Scrum, debe terminarse el anterior y esto determina si se completó el objetivo y si la duración entre cada evento es la idónea.

La implementación de Scrum posibilita la toma de decisiones de forma más rápida, empodera a las personas para que actúen por sí mismas [18] y no solo con el equipo, sino también con el dueño y los clientes. El equipo que implementa Scrum conoce cuáles son sus responsabilidades: el manejo de tiempo entre cada iteración, las tareas por realizar y quién es el responsable de cada una [19].

Los ciclos de DevOps son útiles para marcar sus ciclos de vida, estos indican las necesidades de colaboración y ensayo en cada una de sus fases para su correcta mejora. Existen seis fases repetitivas de un ciclo de vida de DevOps que se representa por las capacidades, procesos y herramientas. Cada una de ellas se encuentra separada, pero en colaboración y comunicación constante del equipo para garantizar la velocidad, alineación y la calidad del proceso. Los ciclos de vida son: planificación, construcción, integración y despliegue, monitoreo, operación y respuesta. Es importante que cada equipo esté bien equipado con herramientas y reglas abiertas a cada etapa de desarrollo. [20]

El proceso de planificar se encarga de declarar el problema y el alcance que se debe obtener con el fin de identificar los recursos necesarios que se necesitarán para el desarrollo, herramientas que ayudarán a realizar este proceso son: Atlassian jira *software*, en este se puede desarrollar la planeación mediante flujos de trabajo que son altamente configurables [21]. Azure boards es un servicio que permite la administración de trabajo en proyectos de *software* [3]. Así como Asana, para la gestión colaborativa de proyectos y flujos de trabajo [22].

DevOps, en el proceso de la construcción, utiliza *software* que ayuda proporcionando mecanismos capaces para favorecer la creación de entornos de desarrollo. Las herramientas utilizadas para la construcción pueden ser: Apache Maven, herramienta de automatización de compilación, empaquetado y manejo de dependencias, que se utiliza, principalmente, para proyectos Java [23]. NuGet es un administrador de paquetes para el desarrollo en .NET [24] y Git, sistema de control de versiones que mantiene la integridad de los datos [25].

La integración continua se encarga de detectar las posibles problemáticas que se puedan encontrar en un *software* de manera temprana, y permitirá encontrar una solución hacia el problema. Las herramientas que se pueden utilizar son: Flagsmith, facilita la creación y gestión de marcado de características en aplicaciones de servidor, web y móviles [31]. Jira ayuda a los equipos a entregar valor a los clientes, al lanzarlos antes, con más frecuencia y de manera más iterativa. Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Mercurial y Git [32].

Los despliegues son grandes objetivos de DevOps, por lo que, en este paso, se facilita el paso del *software* que se desarrolla a una versión funcional. Herramientas para esta fase son: Jenkins, permite compilar y probar proyectos de *software* de manera continua. Octopus Deploy es una herramienta automatizada de administración de implementación y lanzamiento [38]; AWS brinda servicios que ayudan a implementar los despliegues continuos en las buenas prácticas de DevOps [39]. Así como Google Cloud Deploy que es una herramienta que permite la entrega continua de una manera eficaz y fácil [26].

La operación en las versiones de *software* funcional permite realizar una monitorización del rendimiento del *software*, donde se pueden obtener datos de problemas y dar el seguimiento correcto de los problemas, incidentes y cambios. Herramientas útiles en esta fase son: Kubernetes, que facilita la automatización y elimina muchos de los procesos manuales durante la implementación y escalabilidad de las aplicaciones en contenedores [27]. Chef permite automatizar procesos y tareas en numerosos servidores y otros dispositivos de una organización en sencillos pasos [28]. Además, Puppet es una herramienta de gestión de sistemas que ayuda a automatizar y centralizar el proceso de gestión de la configuración [30].

### III. METODOLOGÍA

Este artículo aborda la investigación desde un enfoque cualitativo, basado en los lineamientos presentados por Hernández Samperi [33] sobre el proceso de investigación cualitativa explicativa, generando un modelo de análisis basado en hechos anteriores y literatura existente, tomando en cuenta estudios e implementaciones previas. El objetivo principal de la investigación cualitativa explicativa es alcanzar el nivel más alto de profundidad, con el propósito de abordar las causas a partir de las cuales se presenta algún determinado fenómeno, hecho o suceso.

La información utilizada en esta investigación se obtiene de diferentes fuentes, como lo son artículos académicos, bibliográficos e internet; además, se hará un análisis de casos de éxito de compañías importantes en diferentes sectores económicos, y se implementa un pequeño ciclo de DevOps, con el fin de observar y recopilar datos mediante la experiencia propia. Para ello, se selecciona un método de investigación inductivo, mediante el cual se pretende, a partir del análisis de la literatura e implementación, observar los hechos y generar la evidencia necesaria para apoyar las hipótesis definidas durante el proceso de investigación.

Como herramienta de investigación, se hace uso de la observación sistemática, la cual permite recolectar datos con parámetros ya definidos, en este caso, el uso de DevOps y metodologías ágiles, mediante el muestreo de acontecimientos relacionados a la implementación de ambos marcos de trabajo en las organizaciones y el efecto que se produjo en ellas post implementación.

El análisis sigue las etapas mencionadas en el libro de Hernández Samperi [33]. Una vez recolectados los datos, se realiza el análisis de estos mediante un cuadro comparativo y la validación de las hipótesis, contra los hechos que se evidencian en los resultados del caso práctico implementado y la literatura analizada; los cuales apoyan de manera satisfactoria las hipótesis y validan las preguntas de investigación planteadas con anterioridad.

### IV. ANÁLISIS DE RESULTADOS

Para comprender mejor las distintas fases de DevOps, debe analizarse cómo es un ciclo de desarrollo de *software* sin la implementación de las fases de este. El ejemplo que se describe, a continuación, fue generado con base en la experiencia profesional de varios desarrolladores. Los tiempos mencionados son solo una estimación general; estos pueden variar en cada empresa o en cada programador. Seguidamente, se explicará el escenario: se contrata a un equipo de desarrollo para generar un sistema transaccional, debido a lo grande del sistema, se divide en diversos entregables y se espera que cada entregable funcione correctamente.

El primer entregable consiste en generar la página de inicio de sesión (login), la cual debe verificar si el usuario existe en el sistema; caso contrario, indicar, por medio de un mensaje, que el usuario no existe. En las siguientes tablas, se observan los tiempos por días de cada una de las fases que lo componen, si se emplea DevOps con Scrum o no.

Escenario			
Etapa	Sin DevOps ni Scrum	DevOps con Scrum	Etapa
Planeación	3 días	1 día	Planeación
Desarrollo	15 días	5 días	Construir
Prueba	3 días	1 día	Pruebas automatizadas - Planeación
Solución de errores	7 días		Pruebas automatizadas - Ejecución
Solución de errores en ambientes de producción	7 días	3 días	Integración
<b>Tiempo total</b>	<b>35 días</b>	1 día	Reunión con el cliente y despliegue
		<b>11 días</b>	<b>Tiempo total</b>
		Tiempo indefinido	Monitoreo y retroalimentación

Tabla 1 Comparación de escenarios.

Fuente: Elaboración propia, 2021

El primer aspecto que se ve reflejado es la reducción de tiempos totales. Donde el mismo proyecto tomó 35 días y el segundo escenario tan solo 11 días, se presentó una reducción de 24 días. Para comprenderlo de una mejor manera, se debe analizar cada etapa por separado. El escenario A será sin DevOps y el escenario B con DevOps.

En la etapa de planeación, el escenario A no dispone de buenas prácticas, se realizan los diseños preliminares de la página, bases de datos y las tecnologías por usar. Por otro lado, en el escenario B, se toma que el desarrollo debe durar máximo 15 días, por eso en un solo día se definen los responsables, clasifican las tareas y definen una fecha límite por cada tarea; la cual emplea la herramienta Jira.

En la segunda etapa, la cual es el desarrollo, en el escenario B cada día se realiza una revisión del avance que se posee, la cual dura 15 minutos, además de llevar un control del código con Git; en cambio, en el escenario A solo se programan las funcionalidades por separado. Al finalizar el periodo de desarrollo, se realizan las pruebas en el escenario B; se hace de forma automatizada, donde se vela por el adecuado funcionamiento empleando Selenium; aun así, el escenario A se ejecuta de forma manual, lo que ocasionó que se prolongara el tiempo para la corrección de errores encontrados.

Para el escenario A, los ambientes de pruebas y de producción eran completamente distintos y al ejecutar las pruebas, generó errores en el código; ante esto, se tuvo que corregir los errores en el ambiente de producción. En cambio, el proceso del escenario B fue relativamente sencillo; primero, se realizaron pruebas a un grupo enfocado por medio de Flagsmith; al completar estas pruebas, se conversó con el cliente y aprobó para que se probara al ambiente de producción y no se presentó ningún error.

En el escenario B, debe continuar el ciclo de vida, dado que es un proceso de mejora donde se debe monitorear cómo la aplicación consume los distintos recursos del servidor de producción; además, si surgen nuevas observaciones, se debe mencionar para que el equipo de desarrollo trabaje los nuevos cambios y volver a empezar el ciclo de DevOps.

En un adecuado proceso de DevOps, el equipo de desarrollo incluye los roles normales, además, incluye a la dirección de operaciones y arquitectos para generar una estrategia que considere todo el ciclo de vida del proyecto [36], lo cual se ve reflejado en el escenario B. Por otro lado, se evidencia que, en cada etapa, es un proceso incremental y los tiempos de entrega son más veloces y con una mayor calidad, lo cual es un aspecto esencial de DevOps [37].

Ahora, se debe analizar uno de los casos que ha ayudado a que DevOps sea parte de la ruta de la cultura ágil en las organizaciones a nivel mundial, convirtiéndose en un pilar fundamental de las organizaciones modernas; se hace referencia al caso Netflix.

Cerca del año 2012, la compañía se dio cuenta de que uno de los servicios que ofrecía era muy costoso, ya que los gastos operativos y el ciclo de pruebas de este eran bastante tediosos. La comunicación entre los equipos implicados no era la idónea y, como si fuese poco, estos no asumen sus responsabilidades. Sumado a esto, se recalca que los ciclos de pruebas se realizaban de forma manual y para validar el correcto funcionamiento de cada uno de los cambios que se realizaban a este servicio, se duraba en promedio una semana. La compañía, en búsqueda de solucionar la situación que los aquejaba, en un principio optó por la implementación de un modelo híbrido, el cual no les generó resultados positivos, dado que se analizaron los principios DevOps y determinaron por qué no funcionó la implementación del modelo híbrido inicial.

Lo primero fue analizar el ciclo de vida de desarrollo interno, notando que cada uno de los segmentos del ciclo de vida de desarrollo de *software* eran muy eficientes, pero de manera individual; no obstante, cuando se trataba de verlos como el ciclo de vida completo, el proceso era ineficiente. Por lo cual, decidieron validar los beneficios que les traerían las buenas prácticas de DevOps y apostar por que el área de desarrollo se encargara del mantenimiento del *software*, buscar resolver los problemas de comunicación, además de crear un equipo especializado para la creación de *software*.

Netflix logró reducir el tiempo que tardaba el proceso a días, lo cual fue muy beneficioso, ya que, gracias a esto, pudo dedicar esfuerzos a innovar y estar preparado ante los cambios del mercado; lanzar nuevas características de manera mucho más periódica, mejorar el rendimiento y la calidad de sus servicios y, tal vez, el punto más importante y que ha sido parte de su éxito, la capacidad de escalar fácilmente. Es una plataforma con gran demanda alrededor del mundo y necesita poder escalar fácilmente según la demanda [34].

Si se observa detenidamente el caso anterior, Netflix presentaba uno de los problemas más comunes que presentan las organizaciones que cuentan con un departamento de desarrollo de *software*: los cuellos de botella al momento de liberar nuevas funcionalidades, la ineficiencia de los procesos de desarrollo y pruebas y, sobre todo, la falta de comunicación entre departamentos y equipos de trabajo. Tras la implementación de DevOps, estos problemas fueron solventados, ayudando a la organización a ser más ágil, lo cual, evidentemente, se traduce en crecimiento y ganancias.

Otro caso que se debe tomar en cuenta es la adopción de DevOps por la multinacional Rabobank, quienes son parte del mercado financiero con sede en los Países Bajos. El área de TI de la organización afronta los retos de reducir los tiempos de los ciclos de desarrollo; disminuir los tiempos en los que el servicio no estaba disponible para los clientes a causa de incidentes en producción y, por último, convertirse en una organización de alto rendimiento.

Para afrontar estos desafíos, la organización combinó su

esquema de desarrollo, la implementación de DevOps y la inclusión de una cultura de trabajo ágil en los equipos de trabajo involucrados en las fases de desarrollo, y obtuvo como resultado un equipo que trabaja de manera homogénea, con liderazgo y que inspira a los demás equipos de la organización a adoptar su esquema de trabajo; una respuesta pronta ante cualquier obstáculo o problema que se presente. Por otra parte, el dominio y la visión del producto les ha permitido a los equipos priorizar las necesidades y generar un *backlog* de trabajo, lo que ha permitido que el equipo de DevOps integrara marcos de trabajo ágil como Scrum o Kanban.

Se ha estandarizado la manera de trabajar de los equipos y del departamento de operaciones como tal, se han reducido los tiempos de inactividad debido a incidentes y se ha empoderado a los dueños de producto en la toma de decisiones críticas para la organización y sus sistemas [35].

Para este caso en particular, aparte de los beneficios que aporta DevOps en las organizaciones, se destaca que la integración con Scrum o Kanban ayudó a la organización a adoptar de manera más rápida las buenas prácticas que ofrece DevOps y, consecuentemente, genera mayores beneficios para la organización aprovechando las ventajas de ambos marcos de trabajo; haciendo que su implementación conjunta maximice la entrega de productos de calidad, los cuales cumplen con los requisitos establecidos por las necesidades de los clientes y de la organización.

## V. CONCLUSIONES Y RECOMENDACIONES

Como la teoría lo indica, DevOps es una forma de abordar la cultura de transformación digital, generando mayor valor y capacidad de respuesta para las organizaciones ante los cambios, proporcionando una serie de beneficios que les permiten mantenerse en el mercado. Al realizar el análisis de los resultados, se evidencia que los problemas más comunes de estas organizaciones se solventan implementando DevOps, además, obtienen beneficios que les permiten ser más competitivos; por lo cual se concluye que la implementación de DevOps en las organizaciones solventa los errores típicos que enfrentan estas en la era digital y, sobre todo, nivela la balanza con respecto a sus competidores, ya que otorga la capacidad de crecer, innovar y competir según las demandas del mercado. Les permite establecer una mejor comunicación entre departamentos, lo cual contribuye al crecimiento integral de esta y a la materialización de objetivos.

Dicho lo anterior, se recomienda acoger DevOps no como una alternativa para solventar errores, sino como parte de la cultura organizacional, para que esta le permita a la empresa maximizar el aprovechamiento de sus recursos, aumentar la productividad, la capacidad de respuesta antes los cambios, competitividad y mejorar la calidad de los servicios brindados, lo cual se traduce en ganancias para la organización.

Como se evidencia en la teoría, DevOps emplea en cada una de sus fases escenarios que maneja Scrum, donde se observa

el manejo de equipo, desarrollo del código y las pruebas de este. Donde cada entregable es incremental y al finalizar la etapa, debe hacerse un proceso de retroalimentación. Ante el ejemplo mostrado como los dos casos de éxito, la comunicación con el equipo es clave y disponer de varias herramientas que permitan gestionar el proyecto es fundamental. Sin olvidar que debe tener clara la necesidad que debe solventar.

Se evidencia la manera en que estas dos formas de trabajo se relacionan y que el equipo de desarrollo debe tener un gran compromiso para trabajar autónomamente, dado que incrementa la calidad del *software*, sin mencionar que habrá una reducción en los tiempos de entrega. Donde cada tarea es responsabilidad del equipo y cada etapa es esencial en el proceso, la integración de estos es posible y factible de hacer.

Entre las recomendaciones que debe seguir están realizar las reuniones del equipo, definir las tareas y tiempos por medio de una herramienta. Que a la hora de codificar se maneje en un mismo lugar, debe conocer las pruebas que se realizarán y conversar con el cliente para indicar si el resultado cumple con los requisitos; estos deben integrarse a la solución final y monitorearse de forma continua.

En el ciclo de vida de DevOps, existe variedad de herramientas que se utilizan en conjunto y que permiten lograr una mejor satisfacción de los clientes, por lo cual, para lograrlos se utilizan diferentes herramientas especializadas a lo largo de los ciclos de DevOps. A lo largo de la investigación, se puso en práctica qué tan funcionales son las herramientas y cuál es el impacto que generan, en combinación con las metodologías, en este caso Scrum.

A partir de lo anterior, se alcanzó un balance que genera rendimientos más que aceptables, situación que no ocurría cuando no se utilizaban las diferentes herramientas que permiten aplicar las buenas prácticas de DevOps en conjunto con Scrum. Ciertas ventajas como son las disminuciones en tiempos, costos y la maximización en rendimiento de los procesos de desarrollo y producción. Es recomendable realizar una investigación exhaustiva para encontrar las mejores herramientas que se acoplan a las necesidades de la empresa, por lo cual, si se elige un buen ecosistema por el que las diferentes herramientas tengan una amplia compatibilidad, serán más notables los beneficios de aplicación de esta metodología con Scrum.

Como objetivo de Scrum se encuentra entregar un incremento de producto durante el final de un *sprint* y la única manera de conseguirlo es mediante la integración del negocio, desarrollo y operaciones. Por su parte, DevOps se orienta hacia las operaciones y que los ingenieros trabajen en colaboración para facilitar la puesta de producto a producción. Por lo cual, es posible observar, en el análisis de los casos de éxito de las empresas que aplicaron la combinación, que generan grandes beneficios en la productividad de la empresa, lo cual se refleja en un crecimiento acelerado, donde se ven beneficios

financieros, y puede reducir los costos elevados de algunos de los servicios que presentan. Por lo cual, es posible ver las mejoras con las ventajas obtenidas de la empresa, ya que se maximiza la entrega de los productos cumpliendo con las necesidades de la compañía. Como recomendación, es importante que, a la hora de implementar buenas prácticas, se sigan los buenos lineamientos y elección correcta de herramientas, así como la capacitación constante del grupo de trabajo.

Para emplear DevOps en la organización, debe disponer de un nivel de madurez en ciertas partes del ciclo de vida; aun así, si la empresa emplea Scrum en su proceso, utilizar la metodología DevOps será un proceso natural. Esto se refleja con los casos de éxito que implementaron DevOps y de forma simultánea usaron Scrum dando resultados aceptables. Por lo que se concluye que, si una empresa implementa ambas metodologías en sus procesos de desarrollo, se obtiene un producto más robusto y acorde a las necesidades de los clientes, generando una mayor satisfacción en el programa. Se debe tomar en cuenta que se debe poseer una responsabilidad en cómo los equipos deben trabajar y completar las distintas fases con la mayor calidad posible.

Los términos de DevOps y *agile* son parte del desarrollo de *software*, por lo cual son metodologías de desarrollo de *software*, pero no son lo mismo. Por un lado, DevOps se refiere a unir a los equipos de desarrollo con sistemas y *agile* consiste en una filosofía de trabajo donde los equipos de desarrollo se apoyan. Las empresas que emplean la cultura de DevOps con metodologías ágiles logran planificar tareas de operaciones mejorando a la hora de hacer pases de producción, desplazarse entre centros de datos y llevar a cabo actualizaciones, por lo cual se concluye que DevOps con metodologías ágiles logran acelerar la entrega de *software* como el rendimiento de los equipos de trabajo.

Se recomienda usar ambas metodologías, ya que el enfoque ágil es muy bueno en cuanto a métodos para la organización del trabajo; mientras que DevOps se encarga de promover una cultura más amplia de entrega de *software* de manera rápida y confiable.

## VI. TRABAJOS FUTUROS

Dado los resultados que se obtienen de la actual investigación, los cuales apoyan las hipótesis sobre los beneficios de implementar DevOps en conjunto con metodologías ágiles, se requiere realizar una investigación, con el fin de validar las herramientas disponibles en el mercado y las más usadas por las organizaciones que han implementado DevOps. Esto con el objetivo de identificar cuáles son las que ofrecen mayores beneficios a las organizaciones, tomando en cuenta cuáles les permiten un mayor aprovechamiento de cada una de las etapas del ciclo de DevOps y una fácil integración con las metodologías ágiles y procesos ya preestablecidos en las organizaciones.

El propósito de esta investigación es generar una base de contenido que le sirva a las organizaciones poder definir qué herramientas pueden llegar a ser más convenientes según el tipo de negocio que desempeñen y los procesos que ya se encuentran establecidos en estas, esto facilitará de manera considerable el proceso de análisis de los procesos y la etapa de definición y planificación previa a la implementación de DevOps.

Adicionalmente, se sugiere que esta base de conocimiento se encuentre disponible para el público en general, ya sea en un almacenamiento en la nube o en algún sitio web donde las organizaciones y personas que deseen ampliar sus conocimientos en esta área puedan comparar las herramientas existentes usando como parámetros su modelo de negocios, procesos y demás aristas, que les permitan determinar la herramienta con más afinidad posible para la organización; mediante el uso de cuadros comparativos y gráficos que le permitan de manera sencilla su interpretación y, con base en ello, tomar una decisión.

## VII. REFERENCIAS

- [1] RedHat, «El concepto de DevOps,» RedHat, 2021. [En línea]. Available: <https://www.redhat.com/es/topics/devops>.
- [2] NetApp, «¿Qué es DevOps?,» NetApp, 2021. [En línea]. Available: <https://www.netapp.com/es/devops-solutions/what-is-devops/>.
- [3] Azure Microsoft, «¿Qué es DevOps?,» Azure, 2021. [En línea]. Available: <https://azure.microsoft.com/es-es/overview/what-is-devops/#devops-overview>.
- [4] V. Cuervo, «7 Beneficios del DevOps,» AiT, 23 marzo 2019. [En línea]. Available: <https://www.arquitectoit.com/devops/7-beneficios-devops/>.
- [5] K. Schwaber y J. Sutherland, «La Guía Scrum La Guía Definitiva de Scrum: Las Reglas del Juego,» noviembre 2020. [En línea]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>.
- [6] "Release on demand - scaled agile framework". Scaled Agile Framework. 27 septiembre 2021 <https://www.scaledagileframework.com/release-on-demand/> (accedido el 7 de diciembre de 2021).
- [7] CloudZero, «7 DevOps Best Practices You Should Be Following Now,» 24 mayo 2021. [En línea]. Available: <https://www.cloudzero.com/blog/devops-best-practices>.
- [8] A. Pathak, «Explora las 30 Mejores Herramientas de DevOps a Tener en Cuenta en 2021,» 13 mayo 2021. [En línea]. Available: <https://kinsta.com/es/blog/herramientas-devops/>.
- [9] V. Bucheli. "Lineamientos para la implementación

del modelo CALMS de DevOps en mipymes desarrolladoras de software en el contexto surcolombiano". SciELO Colombia- Scientific Electronic Library Online. [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&mpid=S1794-192X2020000100081](http://www.scielo.org.co/scielo.php?script=sci_arttext&mpid=S1794-192X2020000100081) (accedido el 30 de octubre de 2021).

[10] "State of devops report 2021 | puppet". Powerful infrastructure automation and delivery | Puppet. <https://puppet.com/resources/report/2021-state-of-devops-report> (accedido el 30 de octubre de 2021).

[11] Beneficios de implementar DevOps en tu empresa. (2020, 11 de agosto). Three Points. <https://www.threepoints.com/int/beneficios-de-implementar-devops-en-tu-empresa>

[12] Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. Agile to DevOps: Smart Skills and Collaborations. Information Systems Frontiers, 22(4), 927–945. 2020. Available at: <https://doi.org/10.1007/s10796-019-09905-1>

[13] Amazon Web Services, "Practicing Continuous Integration and Continuous Delivery on AWS", *D1.awsstatic.com*, 2021. [Online]. Available: <https://d1.awsstatic.com/whitepapers/DevOps/practicing-continuous-integration-continuous-delivery-on-AWS.pdf>

[14] Atlassian. *Qué es el control de versiones / Atlassian Git Tutorial*. [online] Atlassian. 2021. Available at: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>

[15] Rossel, S. Continuous Integration, Delivery, and Deployment. 2017. Packt Publishing.

[16] Yogesh Raheja, Giuseppe Borgese, & Nathaniel Felsen. Effective DevOps with AWS: Implement Continuous Delivery and Integration in the AWS Environment, 2nd Edition: Vol. Second edition. 2018. Packt Publishing.

[17] Cababie, P., & Troilo, F. Metodologías ágiles en equipos de operaciones del área de tecnología de la información (TI). 2021. Documentos de Trabajo, 783, 1–52.

[18] Hadida, S., & Troilo, F. La agilidad en las organizaciones: trabajo comparativo entre metodologías ágiles y de cascada en un contexto de ambigüedad y transformación digital. 2020. Documentos de Trabajo, 756, 1–43.

[19] Gaete, R. Villarroe, I. Figueroa, H. Cornide-Reyes and R. Muñoz, "Enfoque de aplicación ágil con Serum, Lean y Kanban", *Ingeniare. Revista chilena de ingeniería*, vol. 29, no. 1, pp. 141-157, 2021. Available: <https://scielo.conicyt.cl/pdf/ingeniare/v29n1/0718-3305-ingeniare-29-01-141.pdf>.

[20] Vizard, M. and Jarrett, T., 2021. The DevOps Life Cycle. [online] DevOps.com. Available at: <https://devops.com/the-devops-life-cycle> [Accessed 30 October 2021].

[21] Atlassian. 2021. Jira | Issue & Project Tracking Software | Atlassian. [online] Available at: <https://www.atlassian.com/software/jira> [Accessed 30 October 2021].

[22] "Azure Boards | Microsoft Azure". Cloud Computing Services | Microsoft Azure. <https://azure.microsoft.com/en-us/services/devops/boards/> (accedido el 18 de noviembre de 2021).

[23] Asana. 2021. Gestiona en línea el trabajo, los proyectos y las tareas de tu equipo • Asana. [online] Available at: <https://asana.com/es> [Accessed 30 October 2021].

[24] Porter, B., Zyl, J. and Lamy, O., 2021. Maven – Welcome to Apache Maven. [online] Maven.apache.org. Available at: <http://maven.apache.org/> [Accessed 30 October 2021].

[25] Nuget.org. 2021. NuGet Gallery | Home. [online] Available at: <https://www.nuget.org/> [Accessed 30 October 2021].

[26] Google Cloud Deploy - Fully Managed Continuous Delivery. 2021. Google Cloud. <https://cloud.google.com/deploy>.

[27] Orquestación de contenedores para producción. 2021. Kubernetes. <https://kubernetes.io/es/>

[28] *Chef Software DevOps Automation Solutions / Chef*. 2021. Chef Software. <https://www.chef.io>

[29] Partner.microsoft.com. 2021. Microsoft System Center. [online] Available at: <https://partner.microsoft.com/en-us/solutions/microsoft-system-center> [Accessed 30 October 2021].

[30] DevOps | Puppet. 2021. Powerful infrastructure automation and delivery | Puppet. <https://puppet.com/devops/>

[31] Amazon Web Services, Inc. 2021. ¿Qué es la entrega continua? – Amazon Web Services. [online] Available at: <https://aws.amazon.com/es/devops/continuous-delivery/> [Accessed 30 October 2021].

[32] Flagsmith.com. 2021. [online] Available at: [https://flagsmith.com/?utm\\_source=capterra](https://flagsmith.com/?utm_source=capterra) [Accessed 30 October 2021].

[33] Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta. McGraw Hill Education, 2018.

[34] *DevOps: Qué es y el caso de Netflix - quantion digital factory*. (2020, 21 de diciembre). Quantion Digital Factory. <https://www.quantion.com/es/2020/12/21/devops-que-es-y-el-caso-de-netflix/>

[35] Aplicamos DevOps y Agile para mejorar los servicios de TI. 2017. Quint Wellington Redwood. <https://www.quintgroup.com/es-es/casos->

[exito/implementacion-de-devops-para-lograr-una-excelente-entrega-de-ti/](#)

[36] IBM, "DevOps for Dummies", *Ibm.com*, 2021. [Online]. Available: <https://www.ibm.com/downloads/cas/DW04QO54>

[37] P. Rodrigues, J. Macedo, P. França, L. Franz, J. da Silva and J. Cheiran, "DevOps adoption in Junior Enterprise: an experience report of software development", *Sol.sbc.org.br*, 2021. [Online]. Available: <https://sol.sbc.org.br/index.php/eres/article/view/10085/9966>

[38] Complex Deployments Made Easy | Release Management, Deployment & Runbook Automation Tool - Octopus Deploy. 2021. Octopus Deploy. <https://octopus.com/>

[39] AWS | Cloud Computing - Servicios de informática en la nube. 2021. Amazon Web Services, Inc. <https://aws.amazon.com/es>