



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA.

PROCESO DE TITULACION

JUNIO – NOVIEMBRE 2021

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA
PRUEBA PRÁCTICA

INGENIERÍA EN SISTEMAS
PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO (A) EN SISTEMAS

TEMA
ANÁLISIS DE LAS HERRAMIENTAS DEVOPS EN LA INTEGRACIÓN
Y DESPLIEGUE DE PRODUCTOS SOFTWARE

EGRESADO
LEONARDO JAVIER MEDINA RICACHI

TUTOR:
ING. JOSÉ TEODORO MEJÍA VITERI, MSC

AÑO
2021

RESUMEN

La metodología DevOps es una filosofía adoptada en países de primer mundo, la cual está en constante evolución promoviendo un mejor desarrollo de aplicación en el menor tiempo posible, y posibilitando actualizaciones contantes dando una mejoría para los clientes, con la implementación de las herramientas DevOps se promueve un trabajo colaborativo, con mayor fluidez, integración y transparencia entre los equipos que intervienen en el desarrollo de aplicación y las operaciones tecnológicas.

La relación que existe entre las fases del ciclo de vida de Software es cada vez mas estrecha, desde la planificación inicial, pasando por las fases de codificación, compilación, pruebas y publicación o puesta en marcha del software, por lo cual la utilización de las herramientas DevOps integra todas las fases y su posterior supervisión y actualización continua, lo que permite estar a la vanguardia.

El análisis que se desarrollará permitirá seleccionar las herramientas DevOps en cada fase del ciclo de vida en el desarrollo del software acorde a los requerimientos de las empresas que buscan estar entre las primeras con el nivel mas alto en rendimiento, por ello la selección adecuada de herramientas permitiría una solución rápida y eficaz contando con un alto rendimiento en el cliente final.

El uso de esta nueva tecnología permite el avance a mayor velocidad en la creación de software en los diferentes ámbitos actuales, dando confiabilidad y una mejora continua en los procesos con los clientes, logrando el acoplamiento efectivo en todo el proceso del desarrollo del software.

Palabras clave: Herramientas DevOps, Tecnología, Análisis, Software, integración.

ABSTRACT

The DevOps methodology is a philosophy adopted in first world countries, which is constantly evolving, promoting better application development in the shortest time possible, and enabling constant updates, providing an improvement for customers, with the implementation of DevOps tools, collaborative work is promoted, with greater fluidity, integration and transparency between the teams involved in application development and technological operations.

The relationship that exists between the phases of the Software life cycle is increasingly close, from the initial planning, through the phases of coding, compilation, testing and publication or commissioning of the software, for which the use of the DevOps tools integrate all the phases and their subsequent monitoring and continuous updating, allowing you to be at the forefront.

The analysis that will be developed will allow the selection of DevOps tools in each phase of the life cycle in software development according to the requirements of companies that seek to be among the first with the highest level of performance, therefore the appropriate selection of tools it would allow a fast and efficient solution with high performance in the end customer.

The use of this new technology allows for faster progress in the creation of software in the different current areas, giving reliability and continuous improvement in processes with customers, achieving effective coupling throughout the software development process.

Keywords: DevOps tools, Technology, Analysis, Software, integration.

INTRODUCCIÓN

El uso de las herramientas DEVOPS en los proyectos de TI en la actualidad es el claro ejemplo de evolución en el desarrollo de software mediante métodos ágiles para el aumento de la velocidad en la consolidación de los mismos, en nuestro medio aún es un desafío comprender el uso de las herramientas DEVOPS, en el desarrollo de software, definido así por sus siglas en inglés development (desarrollo) y operations (operaciones), combinando personas, procesos y tecnología para ofrecer un valor agregado a los clientes.

En el desarrollo del presente caso de estudio se dará a conocer los principales y mejores conceptos fundamentales para la aplicación de herramientas DEVOPS en el desarrollo de software, aplicando los grandes avances en la cultura y las prácticas ágiles durante la última década donde se expuso el requisito fundamental de un enfoque más holístico del ciclo de vida de la entrega de software.

Cuando se emplean estas herramientas en el desarrollo de software se consigue una capacidad de respuesta más rápida, robusta y eficaz en cuanto a las necesidades de los clientes que requieren de un software a medida y que requieren un producto de calidad, logrando los objetivos empresariales con mayor rapidez y solvencia, los roles como el desarrollo, las operaciones de tecnología de información, ingeniería de calidad y la seguridad del sistema, dentro del ciclo de elaboración y entrega del software se integran con esta nueva metodología, creando una mejor coordinación, produciendo mejores productos con objetivos iguales y eficaces para las empresas.

Esta metodología juega un papel importante en los procesos comerciales entre las empresas desarrolladoras de software y los clientes, desde la planificación de proyectos hasta la entrega del software, por ello la tecnología DEVOPS crece rápidamente en los grandes mercados sobresaliendo entre todas las metodologías actuales, por ello elegir las herramientas adecuadas para la automatización de un software es una ardua tarea.

DESARROLLO

Las empresas hoy en día requieren de la implementación de nuevas aplicaciones o sistemas integrados orientados a lo que exige la sociedad moderna o en su defecto la modificación de los sistemas ya existentes, en los cuales la velocidad es el valor primordial en el desarrollo de los sistemas, sin perder su calidad y eficiencia, mejorando su competitiva y relevancia dentro de un mercado digital que cada día toma mayor protagonismo en los procesos de todas las actividades en la sociedad.

Las practicas modernas requieren de un acoplamiento entre los procesos, técnica y herramientas, proceso que en la mayoría de empresa no existe o han implementado prácticas para mejorar el ciclo de vida del desarrollo del software, entre las cuales se pueden mencionar SCRUM, XP, SAFe, entre otras, las cuales solo ayudan o dan velocidad en ciertos aspectos del ciclo de vida del software, mientras que la metodología DEVOPS con sus herramientas tiene una reducción de tiempo y eficacia en los tiempos de entregas de software.

La falta de la implementación de la metodología DevOps en el desarrollo de software hace que las empresas queden rezagadas en cuanto al mercado digital que busca el desarrollo ágil, veloz, con una mejor calidad, y con la rapidez necesaria para corregir los errores que puedan presentarse, por tanto, dichas empresas renuentes al avance tecnológico como la metodología DevOps tendrán poca viabilidad en el desarrollo de aplicaciones.

Los niveles de eficiencia en el menor tiempo posible en el desarrollo de software son unos puntos de consideración, y probablemente la principal ventaja sobre el ciclo de vida tradicional del software, manteniendo la calidad y seguridad, agregando una mejora continua en todos los procesos.

Tomando en cuenta que “El ciclo de vida de las aplicaciones, el cual inicia con una idea o solicitud del mercado y finaliza una vez el software está liberado en producción, funcionando correctamente y a disposición de sus usuarios se divide en dos partes: una

etapa de diseño y desarrollo, y una etapa de entrega del producto.” (Forsgren, Humble, & Kim, 2018)

Pero no solo basta con aligerar el proceso de entrega de software, también es necesario asegurar que el software es capaz de entregar resultados confiables y predecibles, sin dejar de impactar el negocio, como se acaba de mencionar. Se precisa entonces que los procesos de Ingeniería del Software incorporen prácticas para crear o modificar aplicaciones de software con los criterios que se han venido enunciando. En una economía movida por negocios digitales, donde el software es un actor fundamental para las interacciones entre los clientes y los proveedores de bienes y servicios es imperativo que el despliegue en producción de una nueva versión de una aplicación sea un tema que se pueda resolver en minutos, un par de horas a lo sumo, en lugar de meses o años.

Los sistemas informáticos tienen un rol importante en la actualidad en cuanto a la prestación de servicios dentro de las empresas de cualquier índole, producto o servicio. A medida que las empresas van adquiriendo protagonismo, requiere mayor calidad y funcionalidad en los servicios dentro del software, por tanto, aumenta la seguridad, adaptación a nuevos sucesos y requerimientos.

En el desarrollo de un proyecto de software se requiere sincronización entre varios componentes: requerimientos, metodología de desarrollo, desarrolladores/operadores, infraestructura, servicio a los usuarios. (Belalcázar, 2017)

Las estrategias de alineamiento entre desarrolladores y operadores ante requerimientos de desarrollo de aplicaciones, deben ser coordinadas desde las etapas tempranas de planificación del proyecto. Cuando se opta, además por metodologías ágiles de desarrollo estas estrategias deben aprovechar el dinamismo que habilitan este tipo de tecnologías. (Molina & Rosales, 2010)

Como se enuncia las herramientas DevOps surgen como una respuesta a las exigencias de las nuevas tendencias en la integración de las áreas de desarrollo y

operaciones de Tic's con el propósito de optimizar y mejorar la entrega de servicios dentro del contexto de metodologías ágiles en el desarrollo de software.

El objetivo primordial de DevOps es un trabajo colaborativo donde se pueda mantener la actualización en los sistemas de producción con un rendimiento estable en todo momento dando continuidad a los procesos generados. Los conceptos subyacentes abarcados por DevOps, representan una reformulación sísmica en el contexto de la producción y el soporte de software. En lugar de mantener aplicaciones de ingeniería discretas ("Dev") y gestión de TI ("Ops"), DevOps dicta el uso de equipos más pequeños con experiencia multifuncional para mejorar la funcionalidad del software y los procesos utilizados para entregarlo. (Ravichandran, 2012)

DevOps y Lean son dos pensamientos bastante compatibles entre sí, al analizar a detalle los principios de Lean se puede observar cuando se habla de eliminación de desperdicios visto desde una perspectiva DevOps y el desarrollo de software, desperdicio es aquello que no aporta valor, como código parcialmente terminado o bugs, pero sin duda uno de los mayores desperdicios en el ciclo de vida del software es el movimiento de trabajo entre equipos. (Sutil, 2017)

En DevOps es esta separación la que también se quiere tratar ya que, si se logra la unión entre Dev y Ops, este desperdicio generado en el paso de información, tareas y/o códigos entre los departamentos de Desarrollo y Operaciones de TI, no se evidenciaría o bajaría sustancialmente; de igual manera los tiempos desperdiciados en toma de decisiones por separado se reducirían drásticamente al trabajar estos 2 departamentos en conjunto. (Sutil, 2017)

HERRAMIENTAS DE DEVOPS

DevOps fomenta la integración entre los equipos de desarrolladores y los profesionales de Operaciones de TI, tendiendo a lograr un ciclo iterativo que busque siempre la mejora continua y la entrega de valor al negocio. DevOps tiene su propio ciclo de vida en relación con la automatización del proceso de desarrollo de software desde su concepción (idea) hasta su operación (producto en producción) (Villegas, 2019), el cual en general está compuesto por 8 etapas, aunque estas pueden variar, ya sea porque se simplifiquen etapas o se agreguen componentes de cada una de ellas (Quijano, 2018)

Según los autores citados definen un grupo primordial de herramientas DevOps para el desarrollo orientado con metodologías ágiles y eficaces en el desarrollo del software, entre las cuales podemos mencionar.

Manejo del código fuente

Control de versiones, como también se lo conoce, es una buena práctica de seguimiento de cambios en el código de un software, ya sea para el desarrollo de una aplicación sencilla que requiera el esfuerzo de una sola persona, como para el desarrollo de un proyecto de software a gran escala que involucre el trabajo de un equipo. (Amazon Web Services, 2020)

Integración continua

La integración continua de forma simple es la combinación del trabajo individual de varios desarrolladores al código raíz, cada uno trabaja sus distintas partes de su código y una vez terminado lo integra al código principal para ser compilado, con esto cada vez que se combina una nueva parte de código se realiza una serie de pruebas que permiten detectar errores y solucionarlos a tiempo, la entrega continua por otra parte se asegura de lograr que el software esté preparado para su lanzamiento en cualquier momento, finalmente la implementación continua se encarga que los cambios realizados al software se apliquen de forma automática sin afectar el funcionamiento. (Andersson, 2017)

Virtualización

La integración continua en la cultura DevOps requiere de un testeo regular del código desarrollado día a día, bajo este contexto, la virtualización de servicios resulta bastante necesario ya que proporciona un entorno ininterrumpido para las pruebas (Palacios, 2020), este entorno simula ambientes que cumplan la mismas características y requisitos que los entornos donde será instalado el software, es decir, permite a los desarrolladores y testers una réplica del entorno final, este ambiente permite habilitar todas las fases de prueba previas a las pruebas de usuario final (Israel, 2019).

Cloud

La agilidad y comodidad para gestionar infraestructura y la adaptabilidad de las organizaciones son dos aspectos fundamentales de DevOps, los cuales se pueden alcanzar gracias a Cloud Computing, en el presente ya no solo se considera que el único beneficio de Cloud es un ahorro de carácter económico, su principal beneficio es la facilidad y rapidez con que la Nube aprovisiona y mantiene la infraestructura de TI para ejecución del código fuente y la continuidad de la operación, siendo los beneficios la reducción de costos y el aumento de la competitividad (Arsys, 2017).

CADENA DE HERRAMIENTAS DE DEVOPS

Los seguidores de las prácticas de DevOps a menudo incorporan a su «cadena de herramientas» de DevOps particular algunas herramientas que se adaptan perfectamente a estos métodos. El objetivo de estas herramientas es tratar de optimizar, acortar y automatizar las diversas etapas del flujo de trabajo de creación de software (o «canalización»). Muchas de estas herramientas también promueven los postulados principales de DevOps, como son la automatización, la colaboración y la integración entre los equipos de desarrollo y operaciones.

A continuación, se ofrece un ejemplo de herramientas que se emplean en las diversas etapas del ciclo de DevOps.

- **Planificación.** En esta fase se definen los requisitos y valores empresariales. Algunas herramientas de muestra son Jira o Git, con las cuales se puede hacer un seguimiento de los problemas conocidos y llevar a cabo la gestión de los proyectos.
- **Codificación.** Esta fase implica el diseño del software y la creación del código. Algunas herramientas de muestra son GitHub, GitLab, Bitbucket o Stash.
- **Compilación.** En esta fase se gestionan las versiones y las compilaciones del software, y se utilizan herramientas automatizadas que ayudan a compilar y crear paquetes de código para publicarlos después para la producción. Se utilizan repositorios de código fuente o repositorios de paquetes que también «empaquetan» la infraestructura que se necesita para el lanzamiento del producto. Algunas herramientas de muestra son Docker, Ansible, Puppet, Chef, Gradle, Maven o JFrog Artifactory.
- **Prueba.** Esta fase incluye la realización de pruebas continuas (manuales o automatizadas) para garantizar la calidad de la programación. Algunas herramientas de muestra son JUnit, Codeception, Selenium, Vagrant, TestNG o BlazeMeter.
- **Puesta en marcha.** En esta fase se emplean herramientas que ayudan a gestionar, coordinar, programar y automatizar las tareas de producción de las versiones de productos. Algunas herramientas de muestra son Puppet, Chef, Ansible, Jenkins, Kubernetes, OpenShift, OpenStack, Docker o Jira.
- **Funcionamiento.** En esta fase se gestiona el software durante su producción. Algunas herramientas de muestra son Ansible, Puppet, PowerShell, Chef, Salt o Otter.
- **Supervisión.** En esta fase se identifica y recopila información sobre problemas que surgen en una versión de software específica que se encuentra en producción. Algunas herramientas de muestra son New Relic, Datadog, Grafana, Wireshark, Splunk, Nagios o Slack. (Vintimilla, 2020)

La información recabada presentada en el caso de estudio esta complementada o citada de la literatura encontrada de diferentes medios y autores los cuales a través de sus experiencias en la implementación de las herramientas DevOps en los diferentes conceptos en el desarrollo de software validan dichos procesos a realizarse.

El caso de estudio está complementado debidamente con la metodología a seguir para encontrar puntos clave que permitan conocer de manera técnica y explícita los parámetros que se desean explicar, así como los debidos métodos y tipos de investigación a utilizarse los cuales determinarán la orientación en la búsqueda de los conocimientos en la elaboración del proyecto.

Tomando en consideración el tipo de proyecto investigativo que se va a desarrollar, el tipo de investigación seleccionado se ajustará a los aspectos y adecuarse para la obtención de la información necesaria y adecuada para la validación del proyecto de investigación. La investigación documental es la más adecuada para la compilación de información necesaria para el estudio y comparación de las diferentes herramientas DevOps que se requieren para implementar la metodología que tiene como consigna primordial la ejecución simultánea de los procesos dentro del desarrollo de software sin bajar la calidad de estos sistemas.

La elaboración del proyecto se apoya con el sustento teórico documental de DevOps, encontrado en las diferentes fuentes bibliográficas de DevOps, y así como los sitios oficiales en portales web como el DevOps Institute, la cual cuenta con la información compartida por los expertos del área con su alta experiencia en proyectos de implementación. Se va a considerar todo tipo de información que se pueda obtener de la literatura de autores que comparten experiencias y conocimientos del uso e implementación de la tecnología DevOps, así como la utilización de las herramientas para el óptimo funcionamiento de un ciclo de producción y entrega de software más dinámico y efectivo.

Con la información que se recolectó sobre experiencias en ámbitos empresariales en el desarrollo de software se trata de dar solución a la mayor cantidad de inquietudes, y fomentar la utilización e implementación de la tecnología DevOps en el desarrollo de los softwares modernos. Se evaluaron varios ensayos realizados por autores citados en el documento para llegar a la elaboración de un cuadro comparativo, donde podemos apreciar las características de cada herramienta que puede ser utilizada en cada etapa de ciclo de vida de software, para lograr la implementación de la metodología DevOps, la cual requiere

la integración simultanea de las tareas o procesos que deben seguirse para la agilidad en el desarrollo del software.

Se realiza la etapa de recolección de la información bibliográfica, revisión de las diferentes investigaciones relacionadas al tema planteado, abarcando la mayor cantidad de rubros o empresas existentes en nuestro medio con la finalidad conocer el manejo y realizar un posterior análisis de las diferentes problemáticas existentes en las empresas.

METODOLOGÍA

TIPO DE INVESTIGACIÓN

Para el desarrollo del presente trabajo de investigación se hará uso del tipo de investigación documental, sujeta al mejoramiento en futuras investigaciones además la naturaleza propia de esta metodología de investigación plantea la recolección de información por parte de expertos de la materia.

La elaboración del proyecto se apoya con el sustento teórico documental de DevOps, encontrado en las diferentes fuentes bibliográficas de DevOps, y así como los sitios oficiales en portales web como el DevOps Institute, la cual cuenta con la información compartida por los expertos del área con su alta experiencia en proyectos de implementación.

Se va a considerar todo tipo de información que se pueda obtener de la literatura de autores que comparten experiencias y conocimientos del uso e implementación de la tecnología DevOps, así como la utilización de las herramientas para el óptimo funcionamiento de un ciclo de producción y entrega de software más dinámico y efectivo.

TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Haciendo alusión a lo descrito por el autor en mención, se revisa la documentación encontrada sobre problemas similares al presentado en el presente proyecto de investigación para la resolución de la mayor cantidad de inquietudes, y fomentar la utilización e implementación de la tecnología DevOps en el desarrollo de los softwares modernos.

PROCESAMIENTO DE LA INFORMACIÓN

Durante esta etapa se procede a recabar información ya existente sobre Transformación Digital, Procesos, Técnicas y Herramientas asociados a los movimientos DevOps y Entrega Continua de Software. Esta información se diferencia en diferentes tipos de fuentes incluyendo sitios web, informes y reportes de firmas de investigación de mercado, revistas, artículos científicos, libros y otros trabajos académicos. Esta investigación documental proporciona una visión sobre el estado del tema o problema elegido en la actualidad.

CONCLUSIONES:

Una vez culminado el estudio de caso en el análisis de las herramientas DevOps en la integración y despliegue de productos software, se pueden determinar las siguientes conclusiones luego de realizar las comparaciones y análisis de las herramientas usadas para esta metodología.

La metodología DevOps es mas que un sistema de trabajo común, se trata de una filosofía que se basa en la automatización de todos los procesos y trabajo en colaboración con los ciclos de vida tradicionales en el desarrollo del software.

Una vez establecido el contexto de la metodología DevOps se realiza un cuadro comparativo de las herramientas mas utilizadas dentro de cada etapa con lo que se puede analizar las herramientas que se adecuen de manera profesional para las exigencias tecnologías actuales.

El uso de las herramientas DevOps adecuadas dentro de la elaboración de software permite aumentar las prestaciones brindadas al cliente y usuario al establecer controles efectivos con el monitoreo de errores evitando que llegue a presentarse averías de consideración, con ello se mantienen niveles excelentes de disponibilidad en las aplicaciones.

El nivel de satisfacción del cliente se basa en la medición clave sobre los estándares del adecuado y acertado manejo de recursos en la elaboración de software por tanto los conceptos y el análisis presentado servirá para una correcta implementación de DevOps en el medio actual.

Bibliografía

- Amazon Web Services, I. (2020). *¿Qué es el control de código fuente? - Amazon Web Services*. New Yersey: Amazon.
- Andersson, L. (2017). *ITIL and DevOps – better together*. AXELOS.
- Arsys, F. (2017). *La estrecha relación entre DevOps y Cloud*. Worksmem.
- Atlassian. (2020). *Software de control de versiones: Descripción general*. Bitbucket.
- Belalcázar, A. (2017). *ARQUITECTURA DE UN DATA CENTER CON HERRAMIENTAS DEVOPS*. La Plata: UNIVERSIDAD NACIONAL DE LA PLATA.
- Castellanos, R. (2019). *Mejores Herramientas de Integración Continua para desarrollo de aplicaciones móviles*. Medium.
- Cohen, N. (2019). *Metodología de la investigación, ¿para qué?: la producción de los datos y los diseños*. Buenos Aires: Teseo.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. Portland: IT Revolution.
- Garcia, A. (2016). *Guía Metodológica para ante Proyectos de Investigación*. Maracaibo: Universidad Experimental Libertador.
- Hüttermann, M. (2012). *DevOps for Developers. Integrate Development and Operations, the Agile Way*. Chicago: Apress.
- Israel, M. (2019). *¿Cuáles son las herramientas DevOps?* Viewnext.
- Molina, A., & Rosales, N. (2010). *Identificación de competencias distintivas en la mejora de procesos*. Maracaibo: Adventure.
- Navarro, F., & Villalobos, N. (2005). *Procesos y Productos en la Investigación Documental*. Maracaibo: Universidad del Zulia.
- Quijano, J. (2018). *El ciclo de DevOps, una guía para iniciarse en las fases que lo componen*. Advanced.
- Ravichandran, N. (2012). *Herramientas DevOps, definición de procesos en el desarrollo de Software*. Caracas: Wilmanvier.

Sutil, E. (2017). *El pensamiento Lean y su influencia en el movimiento DevOps*.
Merida.

Villegas, J. (2019). *HERRAMIENTAS DEVOPS*. Buenos Aires: Arkan.

ANEXOS

Se realiza un cuadro comparativo entre las herramientas que pueden ser utilizadas en las diferentes etapas para el desarrollo del software, de igual se realiza un detalle de cada etapa, lo que se puede o debe realizarse, y a su vez de cada herramienta, tomando en cuenta sus características.

Etapa	Detalle	Herramienta	Descripción	Características
Plan	Punto de partida para crear software, se definen responsables y estiman tiempo. Debe ser posible gestionar la planificación de un proyecto de software Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2	Jira	Herramienta web configurable en metodología de software ágil enfocada en el seguimiento de tareas e identificación de errores	<ul style="list-style-type: none"> -Configuración y seguridad por roles y permisos de usuario -Registro de historias de usuario -Sistema de notificaciones -Permite la compartición y manipulación de archivos
		Trello	Herramienta web con versión gratuita, orientada a concepto Kanban simple y visual	<ul style="list-style-type: none"> -Configurable en 20 idiomas diferentes -Versión gratuita sin límite de usuarios -Integración con herramientas como GitHub, Box, Dropbox, Slack, Harvest y Toggle -Variedad de plugins gratis y pagados -Integración con Jira
		Asana	Herramienta web para la gestión de proyectos que enfocada en la identificación	<ul style="list-style-type: none"> -Permite rastreo de tareas y agregar seguidores -Sistema de recordatorio y notificaciones

			de riesgos y eliminación de obstáculos.	<ul style="list-style-type: none"> -No incluye funciones para llevar control del aspecto financiero de los proyectos -Creación de gráficos de Gantt -Tableros Kanban
		Wrike	Herramienta web altamente personalizable con uso nativo de diagramas Gantt.	<ul style="list-style-type: none"> -Configurable con metodología Scrum -Generación automática de informes del estado y progreso de los proyectos (personalizable) -Etiquetas para mejor control de proyectos, carpetas y tareas -Soporte nativo widget para asociarse con Google drive, Jira, Microsoft OneDrive, Slack, Outlook, Github, entre otros. -API para personalizar integración
		Azure DevOps Boards	Herramienta de gestión de proyectos de Microsoft que combina la planeación de sprints y el seguimiento flexible de los elementos de trabajo	<ul style="list-style-type: none"> -Seguimiento del progreso del proyecto en cada etapa del proceso de desarrollo -Paneles de Scrum integrados -Herramientas de análisis y widgets -Integración con Github -Permite personalizar paneles de datos -Compatible e integrable con Microsoft Teams y

				Slack -Acceso a más de 1000 extensión y posibilidad de creación de extensiones propias
		VersionOne	Herramienta web que soporta Scrum, Kanban, XP, metodologías de desarrollo híbridas y que facilita la planificación y seguimiento de las tareas realizadas por los diferentes equipos involucrados en los proyectos de software	-Mapea el flujo de trabajo en todo el ciclo de vida del software, permitiendo la planificación estratégica hasta la integración continua -Configurable con metodologías ágiles como Scrum, Kanban, XP o metodologías híbridas de herramientas y gestión de proyectos -Vistas de progreso y rendimiento personalizadas -Informes en todos los niveles de la organización -Asignaciones presupuestarias, seguimiento del gasto y pronóstico del presupuesto final
	Fase que comprende las herramientas que permiten la gestión y	Github	Red social basado en Git que actúa como sistema de control de versiones	-Admite flujos de trabajo no lineales distribuidos -Escalable (número de usuarios) -Permite trabajos de desarrollo en paralelo -Compatible con comandos Git -Herramientas CI/CD para facilitar la integración de código -Sistema de notificaciones webhook -Admite rollback

Código	respaldo para el desarrollo del código.			-Usa SHA1 para la identificación de objetos en el repositorio
	Las herramientas por excelencia son el control de versiones Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2	Bitbucket	Sistema de control de versiones con plataforma web de Git con integración con Jira, Trello, Bamboo y Confluencer	<ul style="list-style-type: none"> -Multiplataforma -Compatible con comandos Git -Permite integración con herramientas de CI/CD como Jenkins -Admite wikis por proyecto -Permite feedback de los pull request y cambios por parte de usuarios -Rastreo de cambios -Sistema de notificación webhook -Integraciones personalizadas a través de REST API
		Gitlab	Suite de DevOps con sistema de usuario, repositorios públicos y privados e integración de herramientas que agilizan las etapas del ciclo de vida de desarrollo de software.	<ul style="list-style-type: none"> -Multiplataforma -Compatible con comandos Git -Creación de wikis por proyecto -Integración con herramientas de la suite de DevOps -REST API para integraciones personalizadas -Notificaciones mediante sistema webhook -Suite de DevOps enfocado para trabajo en Dockers.

		Subversion	Sistema de control de versiones de Apache abierto.	<ul style="list-style-type: none"> -Sistema básico -Bloqueo de archivos -Repositorio espejo de solo lectura -Nativo cliente/servidor -API para enlaces a varios lenguajes de programación
	<p>Fase en la que se compila el código para generar un artefacto que pasará a las siguientes fases, comprende:</p> <ul style="list-style-type: none"> -Obtención del código -Integración de dependencias -Compilación -Generador de 	Maven	<p>Aplicación de software libre para gestión y construcción de software basado en java enfocada en el manejo de dependencias.</p>	<ul style="list-style-type: none"> -Repositorio central con gran variedad de librerías -Permite realizar secuencias de pruebas unitarias y de integración -Usa estándares de la industria -Permite integración con herramientas de CI/CD -Generación de reportes de pruebas -Generación de artefactos en diversos formatos (.jar/.war) -Integrada a IDE de java -Multiplataforma
				<ul style="list-style-type: none"> -Depuración compartida para mayor velocidad en resolución de problemas -Compilación inteligente incremental -Soporta java, groovy, Kotlin -Es integrable con IDE de java mediante plugin

Build	<p>artefacto</p> <p>Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 3</p>	Gradle	<p>Software para automatización de build que maneja dependencias. Es el sistema de compilación oficial de Android</p>	<ul style="list-style-type: none"> -Estructura de carpetas y archivos para los proyectos -Genera artefactos en formatos JAR, WAR, EAR -Herramientas para empaquetamiento de código basado en Java Virtual Machine -Publicación de artefactos en repositorios Ivy y Maven -Caché de dependencias de terceros para evitar tráfico de red.
		Jenkins	<p>Servidor de automatización de tareas escrito en java. Es de código abierto y cuenta con 2 cualidades importantes:</p> <ul style="list-style-type: none"> -Creación de Pipeline o cadenas de ensamblaje -Disponibilidad de gran cantidad de plugin para agregar funcionalidades <p>Su versión comercial es Cloudbees Jenkins con calificación de 4.5/5 por</p>	<ul style="list-style-type: none"> -Se puede integrar con herramientas de control de versiones como GitHub, bitbucket o gitlab -Se puede integrar con plataformas de nube (AWS, GPC Oracle Cloud) -Herramienta web -Escalamiento automático -Fácil mantenimiento gracias a herramientas incorporadas para actualizaciones -Alrededor de 1000 complementos -Multiplataforma -Concurrencia gracias a la distribución de tareas en múltiples máquinas

			Gartner.	
		Docker	Herramienta que empaqueta software y sus respectivas dependencias en un contenedor virtual, se considera como una de las estrategias actuales de compilación y despliegue de software a nivel global	<ul style="list-style-type: none"> -Portable -Compatible con diversos lenguajes de programación -Comunidad activa con aportes diarios -Permite integración con herramientas de CI/CD
		Azure DevOps	<p>Suite de DevOps de Microsoft que comprende varias herramientas integradas entre sí:</p> <ul style="list-style-type: none"> -Azure Boards -Azure repos -Azure Pipeline -Azure Test Plans -Azure Artifacts 	<ul style="list-style-type: none"> -Herramienta web -Multiplataforma de modo web -Compatible con comandos Git -Alta integración con toda la suite de Microsoft Azure -Permisos de usuario y rol robustos - Permite compilación de código en cualquier lenguaje -Cuenta con un gran número de plugin para la asociación con múltiples herramientas del mercado.

Test	<p>Fase de pruebas en la que se verifican que cambios aplicados no hayan afectado la funcionalidad del software. Esta etapa se compone por diferentes pruebas que pueden ser automatizadas. Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2 y 3</p>	Junit	Framework especializado en pruebas unitarias	<ul style="list-style-type: none"> -Numerosa documentación disponible -Integración a múltiples componentes y aplicativos de java -Generación de reportes -IDE de java genera pruebas automáticas a través de plugin
		Selenium	Conjunto de artefactos informáticos que ejecuta pruebas sobre navegadores web simulando el comportamiento de un usuario frente a una página web	<ul style="list-style-type: none"> -Permite la creación, modificación ejecución y grabación/reproducción de resultados de pruebas en aplicaciones web -Comandos Selenium orientados para la ejecución de acciones sobre objetos de un navegador -Manejo de puntos de verificación (breakpoints) facilitando la depuración -Especializado en pruebas de regresión -Genera reportes a través de videos simulando el comportamiento de un usuario frente a una aplicación web -Compatible con múltiples lenguajes de desarrollo
				<ul style="list-style-type: none"> -Versatilidad en las pruebas (combinaciones de sistemas operativos, navegadores web, resolución de

		<p>SauceLabs</p>	<p>Framework de pruebas basado en Selenium que posee una plataforma web como servidor de pruebas remotas</p>	<p>pantalla y otras características)</p> <ul style="list-style-type: none"> -Comandos Selenium orientados para la ejecución de acciones sobre objetos de un navegador -Manejo de puntos de verificación (breakpoints) facilitando la depuración -Especializado en pruebas de regresión -Integración con herramientas para CI/CD -Genera reportes a través de videos simulando el comportamiento de un usuario frente a una aplicación web -Compatible con múltiples lenguajes de desarrollo
		<p>Device Farm AWS</p>	<p>Entorno de pruebas para aplicaciones web y móviles</p>	<ul style="list-style-type: none"> -Permite pruebas en diferentes dispositivos en paralelo -Permite realizar pruebas de aplicaciones nativas e híbridas para Android, iOS, y FireOS -Permite acceso remoto de aplicaciones Android y iOS para pruebas interactivas -Generación de reportes y grabaciones de calidad e informativos -Pruebas móviles y web sobre dispositivos reales -Posee un API y SDK compatibles con múltiples

				lenguajes
		Jfrog Artifactory	Repositorio universal integrable con cualquier ecosistema existente, que permite rastrear artefactos desde las fases de desarrollo hasta producción	<ul style="list-style-type: none"> -Admite paquetes de software creados en cualquier lenguaje de programación o tecnología -Admite gestión binaria de extremo a extremo -Plataforma web -Gestión de registros de dockers con alta disponibilidad -Permite la automatización y control de todas las fases del ciclo de vida de DevOps
Release	Fase en la cual ya existe un artefacto o ejecutable producto de las fases anteriores, las herramientas empleadas deben enfocarse en 3 aspectos clave: -Seguridad de acceso	Sonatype Nexus	Gestiona componentes desde el desarrollo hasta la entrega: binarios, contenedores, ensamblajes y productos terminados	<ul style="list-style-type: none"> -Soporte avanzado para el ecosistema Java Virtual Machine (JVM) -Compatible con herramientas como Eclipse, IntelliJ, Hudson, Jenkins, entre otros -Autenticación SAML / SSO para una seguridad mejorada y una experiencia de inicio de sesión único
				<ul style="list-style-type: none"> -Análisis automático de vulnerabilidades de imágenes -Posee una base de datos con registros de paquetes y sistemas operativos con vulnerabilidades, usado para analizar imágenes Docker automáticamente

<p>-Seguridad de permanencia</p> <p>-Alta disponibilidad</p> <p>Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2 y 3</p>	<p>Google Container Registry</p>	<p>Manejador de registro de contenedor de Google</p>	<p>-Control de acceso a repositorios e imágenes mediante políticas</p> <p>-Permite la inserción y extracción de imágenes Docker en repositorios para su uso en sistemas de desarrollo propios o en definiciones dentro del servicio Google Kubernetes Engine</p> <p>-Al ser un servicio no se administra ningún servidor</p>
	<p>Docker Hub</p>	<p>Registro de contenedor docker básico con alta disponibilidad de imágenes que pueden ser usadas como base para aplicaciones</p>	<p>-Control de acceso a repositorio e imágenes mediante usuarios</p> <p>-Permite la inserción y extracción de imágenes Docker en repositorios para su uso en sistemas de desarrollo propios o en definiciones de cualquier entorno o servicios</p> <p>-Al ser un servicio no se administra ningún servidor</p> <p>-Alta disponibilidad de recursos por altas tasas de transferencia</p>
	<p>Google</p>	<p>Servicio de almacenamiento de objetos de alta escalabilidad y alcance mundial similar a</p>	<p>-Encriptación de datos</p> <p>-Control de versiones</p> <p>-Sistema de notificaciones configurables</p> <p>-Logs de auditoría de acceso</p>

		Cloud Storage	Amazon Simple Storage Service	<ul style="list-style-type: none"> -Sistema de seguridad de acceso por autenticación de roles y/o ACL -Ciclo de vida configurable
Deploy	<p>Fase de implementación de la nueva versión obtenida en fases previas, herramientas deben permitir el pase automático a producción considerando sistemas de respaldo y rollback</p> <p>Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 3 y 4</p>	AWS CodeDeploy	Servicio de implementación de componentes capaz de ejecutar tareas que permitan la implementación de aplicaciones en diferentes servicios informáticos	<ul style="list-style-type: none"> -Ideal para implementación y actualización de aplicaciones no dockerizadas, código, funciones AWS Lambda -Consola de administración permite la gestión de las implementaciones mostrando el estado de todas las acciones en un mismo sitio -Es posible detener y revertir implementaciones en caso de errores -Permite aumento de capacidad dinámica para las implementaciones que lo requieran -Permite la ejecución de implementaciones en PC locales de la organización -SDK y API de AWS lo vuelven compatible con herramientas de terceros -Sistema de notificaciones push para el monitoreo del estado de las implementaciones
			Software de automatización que permite el aprovisionamiento de	

		Ansible	infraestructura, gestión de configuración y construcción y despliegue de aplicaciones	<ul style="list-style-type: none"> -Admite aprovisionamiento en la nube -Gran volumen de documentación disponible, así como una comunidad activa -Disponible para varias distribuciones de Linux -No necesita agentes desplegados en instancias finales (solo python en nodos)
		Kubernetes	Plataforma de administración y orquestación de servicios y aplicaciones, similar a un sistema de cluster que permite la administración de redes, almacenamiento, escalamiento, balance de carga y despliegue y destrucción de aplicaciones	<ul style="list-style-type: none"> -Permite el despliegue de todo tipo de aplicación siempre y cuando esté en docker -Despliega Docker y permite la integración con herramientas de CI/CD para la obtención de configuraciones de despliegue -No provee ni obliga a usar sistema o lenguaje de configuración -Soportado por plataformas Cloud
			Servicio de administración de contenedores compatible con	<ul style="list-style-type: none"> -Herramienta a cargo de la administración en servidores -Herramientas de AWS se encargan de auto

		Elastic container Services (ECS)	múltiples servicios de AWS, su fuerte es la escalabilidad, fácil configuración y adaptabilidad de herramientas de terceros.	<p>escalamiento y balanceo de carga</p> <ul style="list-style-type: none"> -Logs y métricas centralizadas en cloudWatch -Sistema de alertas y notificaciones configurable -Integración con herramientas de CI/CD -Proporciona el ambiente idóneo para que una cualquier número de una misma aplicación se pueda ejecutar
	<p>Fase enfocada a los recursos en los que se ejecuta el software. Las herramientas y servicios para esta fase deben facilitar el aprovisionamiento de infraestructura y monitoreo de componentes. Estas herramientas son las sugeridas para seleccionar y evaluar</p>	Vagrant	Herramienta de construcción y administración de Infraestructura centrado en la automatización	<ul style="list-style-type: none"> -Integración con herramientas de administración de configuración como Chef, Puppet, Ansible y Salt -Multiplataforma (Mac, Linux, Windows) -Provisiona un entorno de desarrollo ideal para equipos DevOps
			Servicio de monitoreo para recursos en la nube de AWS y aplicaciones. Permite la visualización de todo el sistema respecto al uso de recursos, rendimiento de aplicaciones,	<ul style="list-style-type: none"> -Configurado para integrarse de forma inmediata con Elastic Compute Cloud -Tableros de control de rendimiento para instancias RDS (Servicio de Bases de Datos relacionales) -Monitoreo de latencia de lectura/escritura y mediciones similares a través de Amazon Elastic block store

Operate	su implementación en el nivel de madurez 3 y 4.	CloudWatch	estado operativo	<ul style="list-style-type: none"> -Monitoreo de métricas como espacio de almacenamiento o memoria libre de instancias de bases de datos RDS -Monitoreo de mensajes enviados/recibidos y métricas similares en colas de mensajes (Amazon SQS) -Monitoreo de mensajes publicados/entregados y métricas comunes en Servicios de notificación de Amazon (AWS SNS)
		Chef	Automatiza la configuración y mantenimiento de servidores de una organización, y es integrable con plataformas basadas en la nube	<ul style="list-style-type: none"> -Fomenta la implementación de la Infraestructura como código -Acelera la adopción de Cloud -Automatiza el proceso de gestión de configuraciones -Actualizaciones de hardware y aplicaciones aplicadas de forma dinámica
		Terraform	Software de aprovisionamiento de infraestructura que soporta varias plataformas Cloud	<ul style="list-style-type: none"> -Permite la administración de recursos y monitoreo del versionamiento de estos -Adaptable a los recursos de la plataforma Cloud donde se ejecuta -Guarda el estado de los recursos administrados -Permite el cálculo de cambios que se efectuarán

				<p>sobre la plataforma previo a su ejecución</p> <ul style="list-style-type: none"> -Fomenta la implementación de la Infraestructura como código
		Prometheus	<p>Base de datos diseñada para para el monitoreo de objetivos, servidores, bases de datos, máquinas virtuales, entre otros</p>	<ul style="list-style-type: none"> -Modelo de datos simple pero potente -Lenguaje de consulta que permite el análisis del rendimiento de aplicaciones e infraestructura -Almacena series temporales en memoria y en disco local -Bibliotecas personalizadas fáciles de implementar
	<p>Fase en la cual se realiza el seguimiento del comportamiento de la aplicación. El monitoreo debe contemplar la validación de componentes o características de las aplicaciones, comportamiento de los componentes en</p>	New Relic	<p>Servicio de rendimiento de aplicaciones web que proporciona un monitoreo de servidor flexible y dinámico permitiendo a los equipos operativos tomar decisiones respecto a sistemas complejos</p>	<ul style="list-style-type: none"> -Visualización de datos de rendimiento de las aplicaciones como tiempo de respuesta del navegador por geografía, transacciones web realizadas en tiempo real, entre otras -Compatible con todos los lenguajes de desarrollo web -Posee un tablero de instrumentos flexibles -Correlaciona la experiencia del usuario final con el rendimiento de las aplicaciones -Conecta el rendimiento de la infraestructura y la aplicación -Herramientas de diagnóstica permiten el análisis de

Monitoreo	<p>producción, procesamiento de datos generados por los usuarios y modo de uso de la aplicación por parte de los usuarios</p> <p>Estas herramientas son las sugeridas para seleccionar y</p>			errores en tiempo real
		Elasticsearch	<p>Base de datos diseñada para almacenar, recuperar y administrar datos orientados a documentos o semiestructurados. Usado por Wikipedia, StackOverflow, Github, entre otros evaluar su implementación en el nivel de madurez 3 y 4</p>	<ul style="list-style-type: none"> -Datos son almacenados en formato JSON -Es escalable hasta petabytes de datos estructurados y no estructurados -Puede ser usado como reemplazo de almacenes de documentos -Mejora del rendimiento de búsqueda a través de la desnormalización - Código abierto -Funciones expuestas como REST API
		Kibana	<p>Herramienta de código abierto para la exploración y visualización de datos que permite el análisis de registros, monitoreo de aplicaciones y casos de uso de inteligencia operativa</p>	<ul style="list-style-type: none"> -Generación de histogramas, gráficos de líneas, gráficos circulares, mapas térmicos y soporte geoespacial -Integración con Elasticsearch -Permite la restricción de datos lo que facilita el intercambio seguro de estos entre paneles -Contiene varios métodos de búsqueda en los datos de registro -Generación de informes.

		Sentry	Plataforma para el monitoreo de errores, permite descubrir, clasificar y priorizar errores en tiempo real.	<ul style="list-style-type: none"> -Mejora de la supervisión del rendimiento de la aplicación con trazas de pila -Diferencia errores identificados con anterioridad de los errores nuevos -Monitoreo de datos en tiempo real
		Datadog	Herramienta de monitoreo y análisis que permite determinar métricas de rendimiento a través de monitoreo de eventos de infraestructura y servicios en la nube.	<ul style="list-style-type: none"> -Permite el monitoreo de servidores, bases de datos y diversas herramientas -Software disponible para implementación on-premise o como servicio (SaaS) -Multiplataforma (Windows/Linux/Mac) -Le brinda al equipo de TI/DevOps una vista única de su infraestructura -Tableros son personalizables -Alertas basadas en problemas críticos -Recopilación y análisis automático de registros, tasas de error, latencias, entre otras

Tabla 1. Comparación de herramientas de la Metodología DevOps

Autor. Leonardo Medina Ricachi