



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA.

PROCESO DE TITULACION

DICIEMBRE – ABRIL 2022

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 HERRAMIENTAS (DevOps) PARA MEDIR SUS
BENEFICIOS EN EL DESARROLLO, INTEGRACIÓN Y DESPLIEGUE
DE PRODUCTOS DE SOFTWARE**

EGRESADO

MARLON CESAR PAUCAR CARPINTERO

TUTOR:

ING. MIGUEL ZUÑIGA

AÑO

2022

TEMA

**ANÁLISIS DE HERRAMIENTAS (DevOps) PARA MEDIR SUS
BENEFICIOS EN EL DESARROLLO, INTEGRACIÓN Y DESPLIEGUE
DE PRODUCTOS DE SOFTWARE**

RESUMEN

El enfoque DevOps es una filosofía adoptada por países del primer mundo, está en constante evolución para promover un mejor desarrollo de aplicaciones en el menor tiempo posible y posibilitando actualizaciones continuas para brindar mejoras a los clientes, la implementación de herramientas DevOps facilita el trabajo colaborativo, con mayor fluidez, integración y transparencia entre los equipos involucrados en el desarrollo de aplicaciones y operaciones técnicas.

Cada vez existe una relación más estrecha entre las fases del ciclo de vida del software, desde la planificación inicial, hasta la codificación, compilación, pruebas y liberación o liberación de software, el uso de las herramientas DevOps integra todas las fases y su posterior seguimiento y control. para estar a la vanguardia.

El análisis que se desarrollará permitirá la de herramientas DevOps en cada fase del ciclo de vida en el desarrollo de software de acuerdo a los requisitos de las empresas que buscan estar entre las primeras con mayor nivel de desempeño, por lo tanto, la adecuada selección de herramientas Permitirá una solución rápida, eficiente y de alto rendimiento en el cliente final.

El uso de esta nueva tecnología puede acelerar el progreso actual de creación de software en diferentes campos, brindar a los clientes confiabilidad en los procesos y mejora continua, y lograr un cierre efectivo en todo el proceso de desarrollo de software.

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

ABSTRACT

The DevOps approach is a philosophy adopted by first world countries, is constantly evolving to promote better application development in the shortest possible time and enabling continuous updates to provide improvements to customers, the implementation of DevOps tools facilitates collaborative work, with greater fluidity, integration and transparency between the teams involved in the development of applications and technical operations.

There is an increasingly closer relationship between the phases of the software life cycle, from initial planning, to coding, compilation, testing and release or software release, the use of DevOps tools integrates all phases and their subsequent monitoring and control. to be at the forefront.

The analysis that will be developed will allow the use of DevOps tools in each phase of the life cycle in software development according to the requirements of companies seeking to be among the first with the highest level of performance, therefore the proper selection of tools will allow a fast, efficient and high-performance solution in the final customer.

The use of this new technology can accelerate the current progress of software creation in different fields, provide customers with process reliability and continuous improvement, and achieve effective closure throughout the software development process.

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

INTRODUCCIÓN

El uso de herramientas DEVOPS en proyectos de TI hoy en día es un claro ejemplo de desarrollo de software aumentando la velocidad de integración a través de métodos ágiles. En nuestro medio, entender el uso de herramientas DEVOPS en el desarrollo de software sigue siendo un reto, definición, por tanto, sus siglas en ingles development (desarrollo) y operations (operaciones), combinando personas, procesos y tecnología para brindar valor agregado a los clientes.

Durante el desarrollo de este caso de estudio, se revelarán los principales y mejores conceptos fundamentales de la aplicación de herramientas DEVOPS en el desarrollo de software, aplicando los grandes avances en cultura y prácticas ágiles de la última década donde se han expuesto los requisitos. Un enfoque holístico del ciclo de vida de la entrega de software.

Cuando estas herramientas se utilizan en el desarrollo de software para lograr una capacidad de respuesta más rápida, más robusta y más eficiente para alcanzar los objetivos comerciales con mayor rapidez y solvencia en función de las necesidades de los clientes que necesitan software personalizado y productos de calidad , los roles como el desarrollo de software crear y dentro el ciclo de entrega, las operaciones de tecnología de la información, la ingeniería de calidad y la seguridad del sistema se combinan con este nuevo enfoque para una mejor coordinación y producir mejores productos para empresas con objetivos iguales y efectivos.

Esta metodología juega un papel importante en los procesos de negocio entre las empresas de desarrollo de software y los clientes, desde la planificación del proyecto hasta la entrega del software, por lo que la tecnología DEVOPS está creciendo

rápidamente en los grandes mercados, destacándose entre todas las metodologías actuales. La automatización del software es una tarea abrumadora.

Este caso de estudio está orientado a la sublínea de investigación de la carrera de ingeniero de sistemas, pues la línea de investigación es “Comunicación y emprendimientos empresariales y tecnológicos, desarrollo de Sistemas informáticos”.

DESARROLLO

Las empresas de hoy necesitan implementar nuevas aplicaciones o sistemas integrados para las necesidades de la sociedad moderna, o modificar los sistemas existentes, donde la velocidad es el principal valor del desarrollo del sistema. Perfeccionar la relevancia y competitividad en el mercado digital manteniendo la mejor calidad y eficiencia, incrementando día a día su desempeño en todas las actividades de la empresa.

Las prácticas modernas requieren del almacenamiento entre procesos, tecnologías y herramientas que no existen en la mayoría de las empresas o han implementado prácticas para mejorar el ciclo de vida del software, las mismas que se son SAFe, SCRUM, XP. Al ayudar o acelerar algunos aspectos del ciclo de vida del software, el método DEVOPS y sus herramientas reducen el tiempo de entrega y la eficiencia del software.

La falta de implementación de metodologías DevOps en el desarrollo de software deja a las empresas rezagadas en el mercado digital buscando agilidad, desarrollo rápido, mejor calidad y la velocidad necesaria para corregir los errores que pueden surgir. Dando como resultado, las empresas que se abstengan a la hora de actualizar sus herramientas tecnológicas, como las metodologías DevOps, las cuales no son muy eficaces en el desarrollo de app.

El nivel más eficaz en el desarrollo del software, se considera una ventaja principal sobre el ciclo de vida del software tradicional, manteniendo la calidad y la seguridad, agregando una mejora continua a todos los procesos.

Según (Soth M. & Casthle P, 2017) considero que “el ciclo de vida de una aplicación, que comienza con una idea o requerimiento en el mercado y finaliza cuando

el software sale a producción, funciona correctamente y está disponible para los usuarios, se divide en dos partes: la fase de diseño y la fase de desarrollo y entrega del producto”.

Pero no solo es suficiente para simplificar el proceso de entrega de software, también es necesario garantizar que el software pueda brindar resultados confiables y predecibles sin dejar de tener un impacto en el negocio, como se mencionó anteriormente. El proceso de ingeniería de software es entonces necesario para incorporar prácticas para crear o modificar aplicaciones de software con los estándares establecidos. En una economía impulsada por los negocios digitales, el software es un actor fundamental en la interacción entre clientes y proveedores de bienes y servicios, por lo que el uso de aplicaciones con las nuevas actualizaciones puede resolver problema en minutos.

Hoy en día, los sistemas informáticos juegan un papel importante en la prestación de servicios a las empresas de cualquier tipo, producto o servicio. En el momento que las empresas ganan popularidad, aumenta la demanda de sus productos por la superioridad en funcionalidad y calidad de los servicios en el software como; seguridad, adaptabilidad, eventos y requerimientos de los sistemas.

Según (Rodriguez C, 2017) en su planteamiento sobre el desarrollo de un proyecto de software se requiere la funcionalidad entre varios componentes: requisitos, metodología de desarrollo, desarrolladores/operadores, infraestructura, servicios a los usuarios. Como mencionan los autores (Disalvo & Muñoz, 2017):

Ante las demandas de desarrollo de aplicaciones, las estrategias de coordinación entre desarrolladores y operadores deben coordinarse desde las primeras etapas de planificación del proyecto. Cuando se elijan, estas estrategias deben

aprovechar el dinamismo que trae esta tecnología, además de métodos ágiles de desarrollo. (pág. 76).

Como se mencionó anteriormente, las herramientas DevOps surgieron en respuesta a las necesidades de las operaciones de las nuevas tendencias en la integración de los dominios de desarrollo y de Tic, con el objetivo de optimizar y mejorar la entrega de servicios en el contexto de un enfoque ágil para el desarrollo de software.

El principal objetivo de DevOps es trabajar en conjunto para que las actualizaciones de los sistemas de producción puedan mantener siempre un rendimiento estable, dando así continuidad al proceso resultante. Según la definición de (Hinojosa D & Pacheco P, 2017):

Los conceptos fundamentales incluidos en DevOps representan una reconfiguración sísmica en el contexto de la producción y el soporte de software. En lugar de mantener la ingeniería de aplicaciones (“Dev”) y la administración de TI (“Ops”) discretas, DevOps requiere el uso de equipos más pequeños con experiencia multifuncional para mejorar la funcionalidad. (pág. 15-35).

DevOps y Lean son dos ideas muy compatibles, al analizar en detalle el principio Lean, desde la perspectiva de DevOps y “el desarrollo de software, podemos ver que el desperdicio es algo que no agrega valor, como el código, la finalización parcial o el error, pero sin duda, uno de los mayores desperdicios en el ciclo de vida del software” es el movimiento de trabajo entre equipos. (Perez D, 2017)

De acuerdo a lo planteado por (Ethan Thorpe, 2017): En DevOps también se espera que se resuelva esta separación, ya que, si se logra la integración entre Dev y Ops, entonces no se comprobará o reducirá sustancialmente este desperdicio de pasar

información, y/o código entre tareas de desarrollo y operaciones de TI; cuando los dos departamentos trabajan juntos, el tiempo perdido en las decisiones individuales se reduce drásticamente.

HERRAMIENTAS DE DEVOPS

DevOps fomenta la integración entre los equipos de desarrolladores y los profesionales de operaciones hacia TI, inclinándose un ciclo iterativo, buscando siempre la mejora continua y entregando valor al negocio. El autor (Jhon AllSpaw, 2017) menciona que:

DevOps tiene su propio ciclo de vida, relacionado con la automatización del proceso de desarrollo de software, desde el concepto (idea) hasta la operación (producto en producción) y se divide en 8 fases, que varían tanto por la adición de componentes por etapas o simplificación de las mismas. (pág. 78).

De acuerdo a lo mencionado en los párrafos anteriores sobre las herramientas DevOps se menciona la importancia de estas herramientas para el desarrollo orientado a la utilización de metodologías ágiles de las que se mencionan.

Gestión de código fuente

Se sabe que el control de versiones es una buena práctica para rastrear cambios en el código del software, ya sea para desarrollar aplicaciones simples que requieren el esfuerzo unipersonal o pluripersonal para el desarrollo de software a gran escala. (Paz Arias A, 2017).

Integración continua

Una forma sencilla de integración continua es combinar el trabajo individual de varios desarrolladores en el código raíz, cada uno trabajando en una parte diferente de su código, y una vez hecho esto, integrarlo en el código principal para compilarlo. Según (Moreira P, 2017) indica que:

Cada vez que haya una nueva parte del código, se lleva a cabo una serie de pruebas para permitir que los errores se encuentren y resuelvan de manera oportuna, la entrega continua, por otro lado, asegura que el software esté listo para su lanzamiento y, finalmente, la implementación continua asegura que los cambios hechos al software se aplican automáticamente sin afectar la operación. (pág. 99).

Cloud

La agilidad y comodidad de administrar adaptabilidad y la infraestructura y la de la organización son dos aspectos fundamentales de DevOps, que se pueden lograr a través de la Cloud Computing, que se considera un beneficio de ahorro económico. Según plantea (Mariño H, 2020):

El principal beneficio es la simplicidad y velocidad con la que la nube proporciona y mantiene la infraestructura de TI para la continuidad de la ejecución del código fuente y las operaciones, con el beneficio de reducir costos y aumentar la competitividad. (pág. 45).

Virtualización

De acuerdo a la investigación de (Pinaloa H, 2017), La integración continua en una cultura DevOps requiere pruebas periódicas del código que se desarrolla a diario, en

este caso la virtualización de servicios es muy necesaria ya que proporciona un entorno de prueba ininterrumpido. Este ambiente simula un entorno con las mismas características y requisitos que aquel en el que está instalado el software, es decir, permite a los desarrolladores y probadores replicar el entorno final que permite habilitar todas las fases de prueba antes de finalizar la prueba de usuario (Mendieta I, 2019).

CADENA DE HERRAMIENTAS DE DEVOPS

Los seguidores de las prácticas de DevOps a menudo agregan a su “cadena de herramientas” específicas de DevOps algunas herramientas que se adaptan bien a estos enfoques. El objetivo de estas herramientas es tratar de simplificar, cortar y automatizar varias etapas del flujo de trabajo de creación de software (o “canalización”). Muchas de estas herramientas promueven los principios básicos, para el desarrollo de operaciones.

Ejemplos de herramientas utilizadas en varias etapas del ciclo DevOps según (Vecillas M, 2020):

Planificación. En esta etapa se definen las necesidades y valores del negocio. Algunas herramientas de ejemplo son Jira o Git, que pueden usar para rastrear problemas conocidos y administrar proyectos.

Codificación. Esta fase consiste en diseñar el software y crear el código. Algunas herramientas de ejemplo son GitHub, GitLab, Bitbucket o Stash.

Compilación. Esta fase administra el lanzamiento y la construcción del software, utilizando herramientas automatizadas para ayudar a compilar y empaquetar el código para su posterior lanzamiento a producción. Utilizando repositorios de código fuente o repositorios de paquetes, también "empaquetan" la infraestructura necesaria para el lanzamiento del producto. Algunas herramientas de ejemplo son Docker, Ansible, Puppet, Chef, Gradle, Maven o JFrog Artifactory.

Prueba. Esta fase incluye pruebas continuas (manuales o automatizadas) para garantizar la calidad de la programación. Algunas herramientas de ejemplo son JUnit, Codeception, Selenium, Vagrant, TestNG o BlazeMeter.

Puesta en marcha. En esta fase, las herramientas se utilizan para ayudar a administrar, coordinar, programar y automatizar las tareas de producción para los lanzamientos de productos. Algunas herramientas de ejemplo son Puppet, Chef, Ansible, Jenkins, Kubernetes, OpenShift, OpenStack, Docker o Jira.

Supervisión. Esta fase identifica y recopila información sobre problemas que ocurren en una versión de software específica en producción. Algunas herramientas de ejemplo son New Relic, Datadog, Grafana, Wireshark, Splunk, Nagios o Slack.

Funcionamiento. En esta etapa, el software se gestiona durante su producción. Algunas herramientas de ejemplo son Ansible, Puppet, PowerShell, Chef, Salt o Otter.

La información recopilada en este caso de estudio es complementada o citada de literatura de diferentes medios y autores que validan el proceso a realizar a través de su experiencia implementando herramientas DevOps en diferentes conceptos de desarrollo de software.

Los estudios de casos complementan adecuadamente la metodología a seguir para encontrar los puntos clave con el fin de comprender de manera técnica e inequívoca los parámetros a explicar, así como el método adecuado y el tipo de estudio a utilizar, lo que determinará la dirección de la busca.

Teniendo en cuenta el tipo de proyecto de investigación a desarrollar, se ajustará el tipo de investigación seleccionado de acuerdo a varios aspectos y se ajustará para obtener la información necesaria y suficiente para validar el proyecto de

investigación. La investigación documental es mejor para compilar la información necesaria para investigar y comparar diferentes herramientas DevOps necesarias para implementar una metodología cuyo mantra principal es ejecutar procesos simultáneamente en el desarrollo de software sin degradar la calidad de estos sistemas.

Se trata de resolver la mayor cantidad de problemas y promover el uso e implementación de técnicas DevOps en el desarrollo de software moderno mediante la recopilación de información sobre la experiencia en el entorno empresarial de desarrollo de software.

Se evaluaron varias pruebas realizadas por los autores citados en el documento, dando como resultado una tabla de comparación donde podemos entender las características de cada herramienta. Esto requiere la integración simultánea de tareas o procesos que se deben seguir para lograr agilidad en el desarrollo de software.

Tipo de investigación

Para el desarrollo de este trabajo de investigación, además de la naturaleza de esta metodología de investigación, se aprovechará la recopilación de información por parte de expertos en el tema, la investigación es de tipo documental.

La elaboración del proyecto se apoya en el soporte teórico de la literatura DevOps (que se puede encontrar en diferentes fuentes bibliográficas de DevOps) así como en la página web oficial de un portal como el DevOps Institute, que cuenta con información compartida por expertos en la materia campo en virtud de su experiencia en la ejecución de proyectos de amplia experiencia.

Se considerará la diversa información disponible de la documentación de autores que comparten experiencia y conocimiento del uso e implementación de técnicas

DevOps, así como la capacidad de usar herramientas para optimizar los ciclos de producción y entrega. y software eficaz.

Técnicas e instrumentos de recolección de datos

En referencia a lo que los autores describieron anteriormente, se revisó la documentación de problemas similares a los presentados en este proyecto de investigación para abordar la mayor cantidad de problemas y fomentar el uso y la implementación de técnicas DevOps en el desarrollo de software moderno.

Comparativa de herramientas DevOps

Se realiza una comparación entre las herramientas en las diferentes etapas del desarrollo de software, detallando así mismo cada etapa, lo que se puede o se debe hacer, y cada herramienta, teniendo en cuenta sus características.

Procesamiento de información

Durante esta fase, se recopiló la información existente sobre transformación digital, procesos, tecnologías y herramientas relacionadas con el movimiento DevOps y la entrega continua de software. Esta información proviene de diferentes tipos de fuentes, incluidos sitios web, informes e informes de empresas de investigación de mercado, revistas, artículos científicos, libros y otros escritos académicos. Este estudio de literatura proporciona una visión general del estado actual del tema o problema seleccionado.

CONCLUSIONES

Una vez que se ha completado el estudio de caso de análisis de herramientas DevOps en la integración e implementación de productos de software, se pueden determinar las siguientes conclusiones después de una comparación y análisis de las herramientas utilizadas para el enfoque.

Una vez establecidos los antecedentes del enfoque DevOps, se elabora una tabla comparativa de las herramientas más utilizadas dentro de cada fase, a través de la cual se pueden analizar las herramientas que se adaptan profesionalmente a las necesidades tecnológicas actuales.

El enfoque DevOps no es solo un sistema general de trabajo, es una filosofía basada en automatizar todos los procesos y trabajar en conjunto con el ciclo de vida tradicional en el desarrollo de software.

El uso de herramientas DevOps adecuado durante el desarrollo de software puede aumentar los beneficios brindados a los clientes y usuarios al establecer controles efectivos y monitorear errores, evitando que prevengan fallas críticas, manteniendo así un excelente nivel de disponibilidad en la aplicación.

El nivel de satisfacción del cliente es una medida clave basada en los criterios para una adecuada y correcta gestión de los recursos en el desarrollo de software, por lo que los conceptos y análisis presentados ayudaron en la correcta implementación de DevOps en el entorno actual.

BIBLIOGRAFÍA

- Carrion E & Palacios J. (2017). Herramientas DevOps, definicion de procesos en el desarrollo de Software. Mexico DF: EDICIONES MARTÍNEZ ROCA.
- Ethan Thorpe. (2017). Influencia en el movimiento DevOps. España: AUSTRAL. Obtenido de https://books.google.com.ec/books/about/Devops.html?id=VqYmygECAAJ&source=kp_book_description&redir_esc=y
- Jhon AllSpaw. (2017). GUÍA COMPLETA PARA PRINCIPIANTES APRENDE DEVOPS PASO A PASO: HERRAMIENTAS DEVOPS. Mexico DF: DEUSTO. Obtenido de https://books.google.com.ec/books?id=ui8hDgAAQBAJ&printsec=frontcover&dq=Devops&hl=es&sa=X&redir_esc=y#v=onepage&q=Devops&f=false
- Jordi Guijarro Olivares, . C. (2020). La estrecha relación entre DevOps y Cloud. Barcelona: Editorial UOC. Obtenido de <https://books.google.com.ec/books?id=hYvcDwAAQBAJ&pg=PT144&dq=La+estrecha+relaci%C3%B3n+entre+DevOps+y+Cloud&hl=es&sa=X&ved=2ahUKewiGsr-RIKn2AhXTQzABHfueDY8Q6AF6BAgEEAI#v=onepage&q=La%20estrecha%20relaci%C3%B3n%20entre%20DevOps%20y%20Cloud&f=false>
- José Juan Mora. (2017). DevOps y el camino de baldosas amarillas. Buenos Aires: EDITORIAL ARIEL. Obtenido de <https://books.google.com.ec/books?id=IdOzCgAAQBAJ&pg=PA284&dq=Devops&hl=es&sa=X&ved=2ahUKewiGylGCKn2AhWURzABHSxLAHUQ6AF6BAgFEAI#v=onepage&q=Devops&f=false>
- Lozaida & Moreno. (2017). Denticion de competencias distintivas en la mejora de procesos. Bogota: EDICIONES PAIDÓS.
- Mendieta I. (2019). ¿Cuáles son las herramientas DevOps? Santiago : Viewnext.
- Paz Arias A. (2017). ¿Qué es el control de código fuente? Madrid: ALIENTA EDITORIAL. Obtenido de https://books.google.com.ec/books?id=uvw0DgAAQBAJ&printsec=frontcover&dq=%C2%BFQu%C3%A9+es+el+control+de+c%C3%B3digo+fuente%3F&hl=es&sa=X&redir_esc=y
- Perez D. (2017). El pensamiento Lean y su influencia en el movimiento DevOps. Barcelona: EDICIONES PAIDÓS.
- Pinaloa H. (2017). ¿Cuáles son las herramientas DevOps? Mexico DF: Austral.
- Rodriguez C. (2017). ARQUITECTURA DE UN DATA CENTER CON HERRAMIENTAS DEVOPS. La Plata: UNIVERSIDAD NACIONAL DE LA PLATA. Barcelona: EL ALEPH EDITORES.
- Soth M. & Casthle P. (2017). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. Lima Ohio: Revolution Zafiro.

Vecillas M. (2020). Herramientas DevOps, definicion de procesos en el desarrollo de Software.
Maracaibo : EDICIONES PAIDÓS.

ANEXOS

Tabla 1. Comparación de herramientas de la Metodología DevOps

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
				<p>Slack</p> <ul style="list-style-type: none"> -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	<ul style="list-style-type: none"> -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 respaldo para el desarrollo del código	Github	Red social basado en Git que actúa como sistema de control de versiones	<ul style="list-style-type: none"> -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
				<ul style="list-style-type: none"> -Usa SHA1 para la identificación de objetos en el repositorio

Código	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.

		<p>Jenkins</p>	<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 Gartner.</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
		<p>Docker</p>	<p>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</p>	<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</p> <p>Estas herramientas</p>	Junit	<p>Framework especializado en pruebas unitarias</p>	<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

	<p>son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 2 y 3</p>	<p>Selenium</p>	<p>Conjunto de artefactos informáticos que ejecuta pruebas sobre navegadores web simulando el comportamiento de un usuario frente a una página web</p>	<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

		SauceLabs	Framework de pruebas basado en Selenium que posee una plataforma web como servidor de pruebas remotas	<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
		Device Farm AWS	Entorno de pruebas para aplicaciones web y móviles	<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. Estas herramientas son las sugeridas para seleccionar y evaluar su implementación en el nivel de madurez 3 y 4.</p>	AWS CodeDeploy	<p>Servicio de implementación de componentes capaz de ejecutar tareas que permitan la implementación de aplicaciones en diferentes servicios informáticos</p>	<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
			<p>Software de automatización que permite el aprovisionamiento de</p>	

		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

		Prometheus	Base de datos diseñada para para el monitoreo de objetivos, servidores, bases de datos, máquinas virtuales, entre otros	<p>sobre la plataforma previo a su ejecución</p> <p>-Fomenta la implementación de la Infraestructura como código</p> <p>-Modelo de datos simple pero potente</p> <p>-Lenguaje de consulta que permite el análisis del rendimiento de aplicaciones e infraestructura</p> <p>-Almacena series temporales en memoria y en disco local</p> <p>-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	New Relic	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>-Visualización de datos de rendimiento de las aplicaciones como tiempo de respuesta del navegador po geografía, transacciones web realizadas en tiempo real, entre otras</p> <p>-Compatible con todos los lenguajes de desarrollo web</p> <p>-Posee un tablero de instrumentos flexibles</p> <p>-Correlaciona la experiencia del usuario final con el rendimiento de las aplicaciones</p> <p>-Conecta el rendimiento de la infraestructura y la aplicación</p> <p>-Herramientas de diagnóstica permiten el análisis de</p>

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 Elastisearch -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

Elaborado por el autor.

Anexo 2:

REPORTE DE URKUND



Document Information

Analyzed document	Marlon_Paucar_Urkund.docx (D131735451)
Submitted	2022-03-28T00:40:00.0000000
Submitted by	
Submitter email	mpaucarc232@utb.edu.ec
Similarity	3%
Analysis address	mzuniga.utb@analysis.orkund.com

Sources included in the report
