

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE ESTUDIOS DE POSGRADO
MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

**MEJORA EN EL PROCESO DE DESPLIEGUE DE APLICACIONES Y MONITOREO DE
FALLAS DEL CENTRO DE INFORMACIÓN, DESARROLLO Y ESTADÍSTICA JUDICIAL
MEDIANTE LA IMPLEMENTACIÓN DE INTEGRACIÓN Y ENTREGA CONTINUA Y
HERRAMIENTAS DEVOPS**

ASESOR: Msc. Ing. Edwin Estuardo Zapeta Gómez
AUTOR: Ing. Josué David Pablo Ixbalán

GUATEMALA, OCTUBRE DE 2020

ÍNDICE GENERAL

ÍNDICE GENERAL	III
INDICE DE ILUSTRACIONES	IV
FIGURAS.....	IV
TABLAS.....	IV
LISTA DE SÍMBOLOS	V
GLOSARIO	VI
INTRODUCCIÓN.....	1
1. ANTECEDENTES.....	2
2. PLANTEAMIENTO DEL PROBLEMA	7
3. OBJETIVOS GENERALES Y ESPECÍFICOS	10
3.1 Objetivo general	10
3.2 Objetivos específicos.....	10
4. JUSTIFICACIÓN.....	11
5. NECESIDADES A CUBRIR Y ESQUEMA DE SOLUCIÓN	12
5.1 Necesidades a cubrir	12
5.2 Esquema de la solución.....	12
6. ALCANCES	16
6.1 Perspectiva investigativa	16
6.2 Perspectiva técnica.....	16
6.3 Perspectiva de resultados.....	17

INDICE DE ILUSTRACIONES

FIGURAS

Figura 1. Proceso de Integración Continua	13
Figura 2. Proceso de Entrega Continua	14
Figura 3. Proceso de monitoreo de fallas	15

TABLAS

LISTA DE SÍMBOLOS

GLOSARIO

INTRODUCCIÓN

1. ANTECEDENTES

En el desarrollo de software, los integrantes de los equipos de desarrollo y operaciones realizan distintas actividades, según AWS en algunas organizaciones, es posible que no existan equipos de desarrollo y operaciones distinto y que los ingenieros se encarguen de todo. (AWS, s.f.)

Los miembros del equipo de desarrollo, programan nuevas características y realizan pruebas de funcionalidad, para verificar la calidad de código. El equipo de operaciones recibe los archivos de código fuente y realizan el despliegue en el ambiente de desarrollo o producción. En la metodología tradicional de desarrollo de software, estos dos equipos están aislados, esto causa algunos inconvenientes como: la retroalimentación de ambos equipos es tardía y las aplicaciones no se implementan de manera correcta, si faltan algunos componentes o paquetes en el entorno de desarrollo o producción. (Kumar AR & TN, 2020)

Las nuevas prácticas y procedimientos de desarrollo de software, han abarcado distintos contextos, tal es el caso de Previed S.A. una empresa de servicios que cuenta con un departamento de TI denominada Apoyo al Giro, este desarrolla la mayor parte de los sistemas de información que apoyan el negocio. Anteriormente el proceso de despliegue de nuevas versiones de los sistemas, se realizaba mediante procesos manuales, los cuales generaban fallos críticos e insatisfacción por parte de los usuarios internos y externos.

En base a la investigación denominada “Mejora del proceso de desarrollo de una empresa de Servicios de Información mediante la incorporación de DevOps” realizada por Díaz Cortés, el objetivo general de este trabajo de tesis fue ajustar los procesos de desarrollo, mediante la implantación de los procesos automatizados de integración y entrega continua, implementando procesos y herramientas de DevOps dentro del departamento de TI de esta organización.

La implementación de estas prácticas y herramientas generó un impacto exitoso, reduciendo los tiempos de despliegue, reducción de errores y automatización de procesos, incrementando la seguridad en el equipo de desarrollo y los usuarios finales. Cabe resaltar que la disminución de daños colaterales después de cada cambio y paso a producción, no depende directamente de las herramientas y prácticas implementadas, sino de la calidad

de análisis y diagnóstico adecuado del problema, esto depende directamente del equipo de desarrollo

Aunque esta investigación no considero la transformación total de la organización mediante la cultura DevOps, se propuso definir un proceso de mejora continua, evaluando los procesos y las plataformas tecnológicas, y realizar los ajustes necesarios para mantener alineado los objetivos estratégicos del departamento de TI. (Díaz, 2019)

Por otra parte, la implementación de integración continua como practica DevOps, se ha vuelto cada vez más común para proyectos de software y pareciera que es la solución a muchos de los problemas a los que se enfrentan los departamentos de TI, sin embargo, que tan útil y beneficioso es su adopción.

Rahman et al., 2017, en su investigación denominada “Continuous Integration: The Silver Bullet?”, concluyen que la integración continua como lo advierte Brooks en el artículo de 1986 llamado “No Silver Bullet - Essence and Accident in Software Engineering”, no es una bala de plata para los desarrollos de software, debido a que los profesionales de la ingeniería, deberán de considerar aplicar las mejores prácticas junto con la adopción de herramientas, así mismo considerar el contexto de desarrollo del equipo antes de la adopción de herramientas, y posteriormente, investigar si satisface las necesidades mediante la aplicación de software analítica. Esto puede ser eficaz para el desarrollo de software, si los integrantes del equipo utilizan el mecanismo de retroalimentación de manera eficiente. (Rahman, Agrawal, Krishna, Sobran, & Menzies, 2017)

El éxito de integración continua (CI, por sus siglas en inglés) se puede acreditar parcialmente a las herramientas, que han automatizado considerablemente la mayoría de pasos para verificar, integrar y probar el cambio del código fuente de manera transparente y directa. Sin embargo, el uso exclusivo de las herramientas de CI, no implica necesariamente que un equipo de desarrollo de software se adhiera adecuadamente a estas prácticas. Investigaciones recientes han demostrado que el uso de herramientas de automatización, puede no producir beneficios, a menos que el equipo de desarrollo esté dispuesto a cambiar su cultura de desarrollo.

Pinto G. et al., 2019, en su investigación denominada “Continuous Integration Theater” resaltan que el ejercicio de estas prácticas puede ser una ilusión de estar practicando la

integración continua mientras que en realidad no se está practicando. Es por ello que los ingenieros de software, deben de ser conscientes de estas malas prácticas que eventualmente pudieran emplear. (Felidré, Furtado, da Costa, Cartaxo, & Pinto, 2019)

La adopción de integración continua ayuda a garantizar que el proceso de desarrollo de software sea confiable, proporciona retroalimentación rápida sobre pruebas y despliegues, reduce la tasa de fallos o errores (bugs) detectándolos rápidamente para que los programadores corrijan estos errores antes del despliegue a producción, ayudan a transformar y a mapear los datos antes de la integración.

Son muchas las ventajas que se pueden destacar respecto a la adopción de integración continua y herramientas DevOps, sin embargo, un detalle muy peculiar surge en la elección de estas herramientas y aún más interesante, alojar estas en la infraestructura local (*On-premise*) o en la nube, esta controversia se convierte en una decisión de gestión crítica para los departamentos de TI.

A medida que la computación en la nube crece y es adoptada por las empresas, sigue siendo la forma más conveniente y rentable de administrar datos extensos. Entonces qué pasa con las empresas que invirtieron grandes cifras en infraestructura local (*On-premise*), generando hasta cierto punto *lock-in* muy fuerte, estos podrían decidirse por alojar estas herramientas de forma local y aprovechar su inversión, sin embargo, podrían no beneficiarse del todo.

Según Makam K. (2020), las herramientas de CI se dividen en distintas categorías: suministrado por el proveedor en las instalaciones, un tercero en las instalaciones, entregado en la nube o herramientas de servicios web API. Estas categorías determinan si los procesos de implementación deben ser plataformas locales o basadas en la nube.

Para una selección efectiva, Makam K, sugiere considerar los siguientes criterios:

- Herramientas de CI que se implementaran: algunas herramientas como *GitHub*, *Heroku* se desenvuelven mejor en la nube, ya que se funcionan con soluciones SaaS.
- Necesidades actuales y futuras de la organización.

- Seguridad de los datos, las organizaciones que están interesados en la seguridad de la información pueden, considerar servidores locales como la mejor alternativa.
- Costos, las soluciones de la nube si bien implican una inversión, se beneficiarán de la falta de gestión de hardware o software esto radica principalmente en el soporte.
- Flexibilidad, personalización y adaptación a las necesidades particulares de la empresa.

La selección de un esquema cloud u on-premise, no depende únicamente de los equipos de desarrollo u operaciones, sino también de la Gerencia de TI, estos deberán de estar alineados a los objetivos estratégicos del negocio. En esta investigación se consideró oportuno, crear una tabla comparativa en donde se requiere de una evaluación de factores como el precio, flexibilidad y la confiabilidad de los proveedores entre otros criterios, que describan las ventajas y la adaptabilidad de cada herramienta para llevar a cabo la selección correcta. (Makam, 2020)

En otra perspectiva, la entrega continua como práctica DevOps, se ha vuelto cada vez más popular y adoptada por gran número de empresas, debido a sus beneficios en el desarrollo de software. Sin embargo, la implementación de entrega continua puede ser desafiante, aunque esta práctica tiene mucha documentación, su adopción sigue siendo un proceso complejo.

Chen (2017) en su investigación denominada “Continuous Delivery: Overcoming adoption challenges” por ayudar a superar los desafíos de adopción, realizó la implementación de entrega continua en una empresa multimillonaria llamada *Paddy Power*, durante cuatro años. En colaboración, logro determinar seis estrategias que pudieran apoyar a este proceso:

- Vender la idea de entrega continua como un analgésico
- Establecer un equipo dedicado con miembros multidisciplinarios
- Entrega continua de la entrega continua
- Comenzar con lo fácil, pero con aplicaciones importantes
- Tuberías de entrega continua
- Expertos en caída

La implementación de entrega continua no depende únicamente de las herramientas, en este proceso se pueden presentar desafíos organizacionales, compatibilidad entre herramientas, legados de sistemas, pruebas de optimización de ejecución y procesos. Es importante que todos los desarrolladores y operadores se involucren, y logren emplear estas estrategias para una adopción exitosa. (Chen, 2017)

En base a los antecedentes planteados, es importante realzar los retos que representan una adopción exitosa, sin embargo, se ha demostrado los múltiples beneficios que estas prácticas y herramientas DevOps han generado, así como estrategias y métodos a emplear. Para el Centro de Información, Desarrollo y Estadística Judicial es importante mejorar los procesos de desarrollo de software y reducir la tasa de fallos en los sistemas, para mantener un nivel de continuidad operativo adecuado, en base a las ventajas de la implementación de integración y entrega continua mediante herramientas DevOps, se espera contribuir a la disminución de la tasa de fallos en los sistemas de información de este centro.

2. PLANTEAMIENTO DEL PROBLEMA

El Centro de Información, Desarrollo y Estadística Judicial (CIDEJ), es la dependencia que, de manera coordinada y consolidándose como única fuente de información, diseña, sistematiza, monitorea y difunde estadísticas, métricas e indicadores judiciales. Así también da continuidad e impulsa la expansión del Sistema de Gestión de Tribunales (SGT) y nuevas aplicaciones tecnológicas. (CIDEJ, 2015)

Entre las principales funciones de CIDEJ, se establece como el principal enlace operativo informático a cargo del diseño, desarrollo, implementación y mantenimiento de los sistemas de información con enfoque jurisdiccional. El repositorio de código fuente alberga más de 10 proyectos, en los cuales la mayoría poseen sus propias copias de trabajo o ambientes de desarrollo.

CIDEJ se enfrenta al problema de que el despliegue de software de nuevas versiones de sus sistemas de información se realiza sin prueba (*testing*) suficiente y mediante un proceso manual que requiere esfuerzo, lo que introduce a errores por su mala ejecución. Si bien la tasa de fallas no es alta (10%), se ha observado que hay mayor demanda por el desarrollo de nuevos servicios y en los últimos dos años se ha observado un crecimiento considerable de Despachos Judiciales.

El despliegue de *software* (*deployment*) de nuevas versiones de los sistemas existentes se realiza con un proceso manual. Consiste en fusionar (*merge*) una o varias características desarrolladas, de un ambiente de desarrollo a un ambiente de producción, utilizando *Tortoise SVN* una herramienta de control de versiones de código abierto. Posteriormente se compilan las nuevas funcionalidades y se remplazan los archivos en un directorio que contiene el proyecto en el servidor de aplicación. Sin embargo, este proceso representa una tasa de fallos considerable, incluso con las mejores intenciones y el mayor cuidado, se producen errores humanos. De acuerdo con las estimaciones realizadas, el 10% de los tickets atendidos en el 2019, se debieron a que los despliegues a producción tenían un error o fallo considerablemente crítico.

Para los Despachos Judiciales, un fallo crítico en producción corresponde a un error en el software, que impide la utilización del sistema, o que afecta la disponibilidad del servicio y

pone en riesgo su continuidad. Estos errores pueden afectar el acceso, la integridad o la consistencia de la información.

De este 10% de errores y fallas, un 6% corresponde a la gestión de calidad en el proceso de desarrollo de software, normalmente por la mala especificación de los requisitos o baja calidad en el análisis. El otro 4% corresponde a errores en el proceso de despliegue en producción de las nuevas características del sistema, como errores en la fusión (*merge*) de archivos de código fuente, mala interpretación de las instrucciones, o configuraciones distintas en cada ambiente. A su vez, al diagnosticar el origen de una falla no tiene la velocidad requerida para permitir una reacción a tiempo.

Un análisis de causas y efectos de estos errores, permitió identificar las siguientes falencias en el proceso de desarrollo y despliegue de software:

1. El nivel de cobertura de *testing* unitario es insuficiente, con apenas un 10% para algunos proyectos, la mayoría de estos test lo realizan los programadores y posteriormente un área de capacitación y atención a los usuarios, que se encarga de verificar las funcionalidades.
2. El nivel de cobertura del *testing* regresivo es mínimo, contando solo un 2% de los proyectos con planificación de pruebas.
3. Por otro lado, se identificó que la retroalimentación de las pruebas funcionales hacia el equipo de desarrollo es tardía, y en muchas ocasiones no se enfocan en las funcionalidades desarrolladas, si no en características que no se describieron en el levantamiento de requerimientos, esto genera controversias y retraso en los tiempos de implementación.
4. En algunos proyectos no existen ambientes de desarrollo, y en otros, los ambientes de desarrollo y producción difieren de características de funcionalidad y configuración. Esta falta de homologación entre ambientes dificulta el diagnóstico de los errores y también ha sido una causa de las fallas.

Otros efectos que puede representar las fallas o errores en el despliegue de software, pueden incurrir en soporte continuo de aplicaciones, recibiendo gran cantidad de tickets y esfuerzo desperdiciado, silos entre equipos y baja confianza en los programadores. Desde la perspectiva de los Despachos Judiciales, la disponibilidad de los servicios puede incurrir

en plazos vencidos, retraso en los procesos, inconsistencia en la información e interferir en la emisión de sentencias y firma de documentos.

Como consecuencia, CIDEJ necesita automatizar el despliegue de sus sistemas a producción, a modo de reducir esta tasa de fallos críticos, y así disminuir los incidentes y mantener un nivel de continuidad operativo adecuado, para los requerimientos y necesidades de los Despachos Judiciales.

La adopción de nuevas prácticas de desarrollo de software ayuda a resolver problemas complejos en estos tiempos de cambio a través de la mejora tecnológica. Con base en lo anterior se plantea la siguiente pregunta central:

¿Qué prácticas y procedimientos deben aplicarse para mejorar el desarrollo de software y reducir la tasa de errores y fallas, en el desarrollo y despliegue de nuevos sistemas de información en ambientes de desarrollo y producción, en CIDEJ?

El proceso de investigación se basa en las siguientes interrogantes auxiliares:

¿Cómo definir una estrategia para la implementación de integración y entrega continua, como aporte a la verificación de la calidad de código, automatización de pruebas, reducir la tasa de fallas y tiempos de despliegue de nuevos sistemas de información en CIDEJ?

¿Qué herramientas permiten la integración y entrega continua, en base a la capacidad de integración, soporte de metodologías ágiles, costos, compatibilidad con sistemas operativos y herramientas de desarrollo de software de CIDEJ?

¿Cómo verificar los resultados de la implementación de integración y entrega continua como prácticas de Desarrollo y Operaciones (*DevOps*), en el desarrollo e implementación de nuevos sistemas de información en CIDEJ?

3. OBJETIVOS GENERALES Y ESPECÍFICOS

3.1 Objetivo general

Implementar un entorno de Integración y Entrega Continua mediante herramientas DevOps, que ayuden a reducir la tasa de fallas, en el desarrollo y despliegue de nuevos sistemas de información en ambientes de desarrollo y producción, en el Centro de Información, Desarrollo y Estadística Judicial.

3.2 Objetivos específicos

Diseñar un plan estratégico para la implementación de integración y entrega continua, como aporte a la verificación de la calidad de código, automatización de pruebas, reducir la tasa de fallas y tiempos de despliegue de nuevos sistemas de información en el Centro de Desarrollo, Información y Estadística Judicial.

Seleccionar e implementar las herramientas DevOps, en base a la capacidad de integración, soporte de metodologías ágiles, costos, compatibilidad con sistemas operativos y herramientas de desarrollo de software del Centro de Información, Desarrollo y Estadística Judicial.

Evaluar el impacto de la implementación de integración y entrega continua como prácticas, de Desarrollo y Operaciones (DevOps), en el desarrollo e implementación de nuevos sistemas de información en el Centro de Información, Desarrollo y Estadística Judicial.

4. JUSTIFICACIÓN

El presente trabajo de graduación corresponde a la línea de investigación de sistemas para impulsar la integración de sistemas de información, enfocándose en el diseño de un plan estratégico para sistemas de información que impactan la calidad del servicio que presta el Centro de Información, Desarrollo y Estadística Judicial (CIDEJ) a los distintos Despachos del Organismo Judicial, por medio de la implementación de un entorno de integración y entrega continua, en base a herramientas DevOps.

Las ventajas de adoptar estas prácticas de desarrollo de software son:

- Entrega rápida y frecuente de software
- Fomenta la cultura colaborativa entre equipos de desarrollo y operaciones
- Automatización de pruebas (testing) y despliegue (deployment) de los sistemas de información
- Alinea los objetivos estratégicos de TI y negocios

Este conjunto de prácticas, ha demostrado ser el método ideal para el desarrollo de software en muchos contextos, debido a que comprende distintas prácticas y herramientas, que mejoran la calidad de código que el equipo de desarrollo escribe, mediante la verificación y retroalimentación temprana de las funcionalidades, y evaluar que cada cambio sea funcional y seguro, beneficiando al equipo de desarrollo en sus entregas de software. La automatización de los procesos de pruebas y despliegue favorece al equipo de operaciones a administrar sistemas complejos y cambiantes de manera eficiente y con un riesgo reducido.

Por lo tanto, el presente proyecto de investigación es un aporte que contribuye con el Centro de información, Desarrollo y Estadística Judicial en el desarrollo de nuevos sistemas de información. Mediante la implementación de estas prácticas, se pretende reducir la tasa de errores y fallas que ocurren principalmente en el proceso de pruebas o despliegue de las aplicaciones.

Como efecto colateral se espera contribuir a la disponibilidad y consistencia de los nuevos sistemas de información de este centro, para que los usuarios finales mediante estas herramientas informáticas sigan ejerciendo sus actividades efectivamente.

5. NECESIDADES A CUBRIR Y ESQUEMA DE SOLUCIÓN

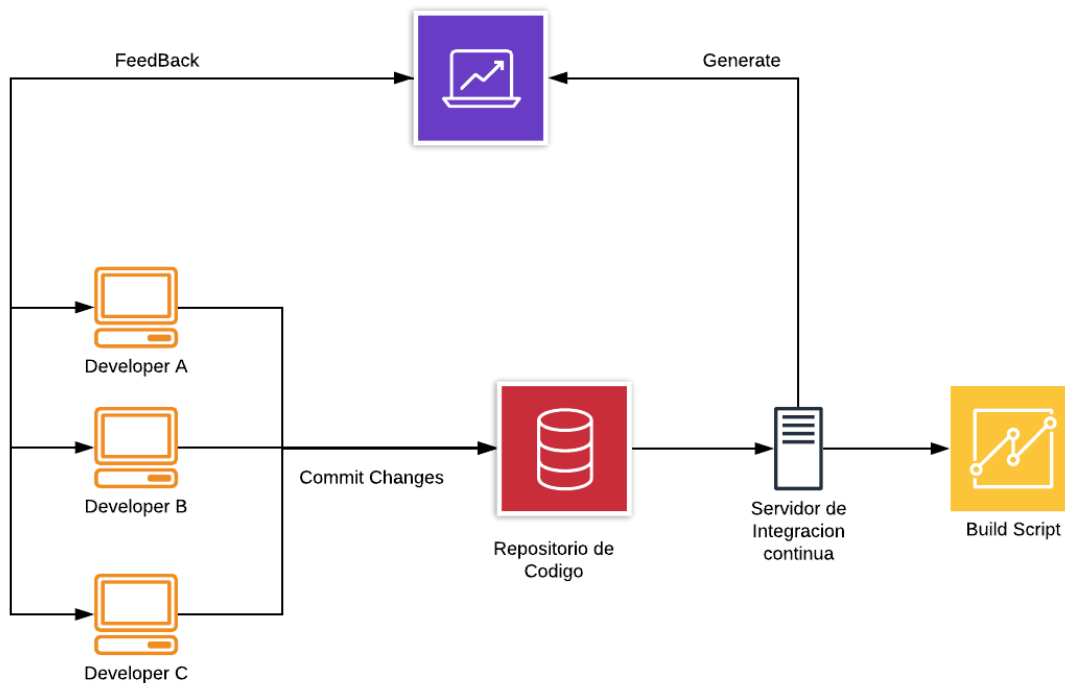
5.1 Necesidades a cubrir

En el entorno de los sistemas informáticos del Centro de Información, Desarrollo y Estadística Judicial, se requiere una respuesta cada vez más rápida y a la vez un servicio que sea estable y seguro. Debido a ello, surge la necesidad de implementar mejores prácticas y herramientas en el ciclo de desarrollo de software, contribuyendo a la mitigación de fallos o errores que se pudieran generar por el despliegue y pruebas de los sistemas, y así mantener un nivel de continuidad operativo adecuado, para los requerimientos y necesidades de los Despachos Judiciales.

Por otra parte, los programadores y operadores necesitan mejorar sus actividades, identificar y mitigar las fallas en el desarrollo de software; con la implementación de integración y entrega continua y herramientas DevOps, se espera agilizar las tareas, fomentar la cultura colaborativa y ayudar a reducir la tasa de fallas, en el desarrollo y despliegue de nuevos sistemas de información en ambientes de desarrollo y producción, en el Centro de Información, Desarrollo y Estadística Judicial.

5.2 Esquema de la solución

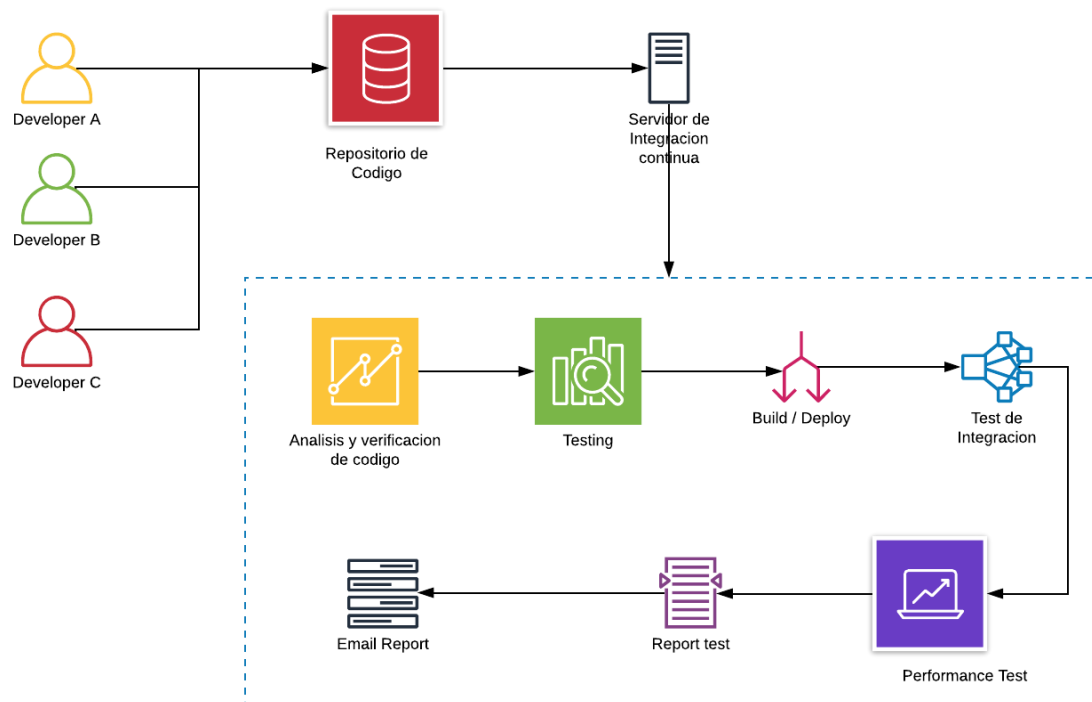
En aspectos generales el proceso de integración continua (Figura 1) contempla los siguientes pasos: los desarrolladores (*developers*) confirman (*commit*) sus cambios de manera continua a un único repositorio de código una o más veces al día, el servidor de integración continua procederá a integrar y a construir los cambios realizados mediante una serie de pasos (*script*), el cual verifica y realiza pruebas unitarias y de cobertura. Generando como resultado retroalimentación efectiva para los desarrolladores.

Figura 1. **Proceso de Integración Continua**

Fuente: Elaboración propia

La implementación de entrega continua, implica generar múltiples ciclos de realimentación para asegurar que el software se entregue con mayor rapidez.

En aspectos generales el proceso de entrega continua (Figura 2) contempla los siguientes pasos: los desarrolladores realizarán confirmaciones (*commit*) de sus cambios en un único repositorio de código fuente, el servidor de control de versiones generará una petición al servidor de integración continua para realizar los procesos correspondientes. A partir de ello se verifica y analiza el código, se realizan test unitarios, cada *commit* debe provocar un *deployment* automático, posteriormente se realizan test de integración, se obtienen los resultados de los test, se generan los reportes y se envían a los desarrolladores con la retroalimentación correspondiente. Como requisito indispensable de la entrega continua, es que todos los *developers* y *operations* colaboren eficazmente en todo el proceso.

Figura 2. **Proceso de Entrega Continua**

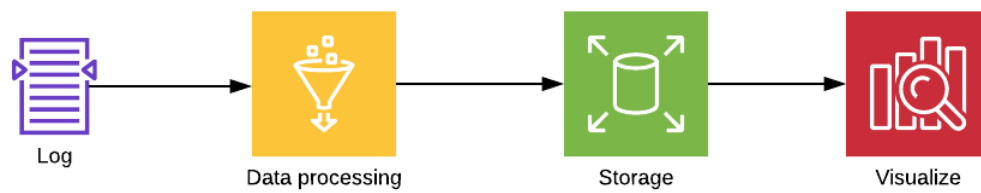
Fuente: Elaboración propia

La monitorización en este proyecto de investigación juega un papel muy importante en la calidad del servicio de los sistemas. Tener un conjunto (set) de pruebas que valide los procesos operativos de un entorno, evitará que las funcionalidades desarrolladas en un sistema de información se ejecuten con errores. Además, monitorizar el estado de la infraestructura, así como de todas las peticiones y operaciones realizadas, mitigará los fallos para que estos no afecten la disponibilidad del servicio.

En general el proceso de monitoreo de fallas (Figura 3) contempla los siguientes pasos: las incidencias que ocurran se detectarán por los servidores web o bitácoras propias del sistema, estos se escriben en ficheros con extensión “.log”, seguidamente se procesan los datos recopilados de una o más fuentes, se filtran y homogenizan para llegar a prescindir de la parte del historial (*log*) que no es importante, seguidamente estos datos son almacenados e indexados en una base de datos, con el fin de tener un historial del

funcionamiento del sistema para extraer métricas y datos de fallas, estos serán visualizados y analizados para generar un plan de mitigación y reducir la tasa de fallos en los sistemas de información.

Figura 3. **Proceso de monitoreo de fallas**



Fuente: Elaboración propia

6. ALCANCES

6.1 Perspectiva investigativa

- Definir un plan estratégico de implementación de integración y entrega continua, evaluando distintos escenarios mediante los sistemas del Centro de Información, Desarrollo y Estadística Judicial.
- Definir buenas prácticas de integración y entrega continua, asociados a una adopción exitosa de herramientas DevOps.
- Describir los conceptos y beneficios de utilizar un entorno de integración y entrega continua, mediante herramientas DevOps y el impacto que estas generan en el proceso de desarrollo de software del Centro de Información, Desarrollo y Estadística Judicial.

6.2 Perspectiva técnica

- Implementar un entorno de integración y entrega continua evaluando los sistemas de información del Centro de Información, Desarrollo y Estadística Judicial que se adapten al esquema de solución propuesto y a las herramientas DevOps seleccionadas.
- Implementar un servidor de integración continua como orquestador principal de las pruebas automatizadas, construcciones (builds) automáticos, despliegues automatizados, resultados y reportes de las pruebas e implementaciones.
- Instalar y configurar herramientas DevOps en los servidores de aplicación e integración continua.
- Implementar y configurar un entorno de monitoreo de fallos basado en un conjunto de herramientas (*stack*)

6.3 Perspectiva de resultados

Implementar un entorno de integración y entrega continua mediante herramientas DevOps en el Centro de Información, Desarrollo y Estadística Judicial que permitan automatizar el proceso de despliegue y pruebas de los nuevos sistemas de información, aportando a la reducción de fallos en ambientes de producción, mediante la detección y monitorización de errores. Las funcionalidades que comprende el esquema de solución son las siguientes:

- Mantener un único repositorio de código fuente
- Analizar y verificar el código fuente desarrollado por los programadores
- Automatización de despliegue (*deploy*) y construcción (*build*) de aplicaciones
- Pruebas unitarias de las nuevas características o mejoras desarrolladas
- Retroalimentación de las pruebas realizadas a los desarrolladores.
- Monitorización de incidencias mediante un tablero (dashboard) y reportes estadísticos.

Como resultados de la implementación de este entorno se puede:

- Definir nuevas estrategias en el proceso de desarrollo de software
- Entregas de software con mayor rapidez y frecuencia
- Optimizar los procesos de despliegue y pruebas de las aplicaciones
- Detección de fallos y errores con mayor rapidez, previas a que se convierten en problemas graves
- Fomentar la cultura colaborativa entre el equipo de desarrollo y operaciones

7. REFERENCIAS BIBLIOGRÁFICAS

AWS. (s.f.). AWS. Recuperado el 21 de 07 de 2020, de <https://aws.amazon.com/es/devops/what-is-devops/>

Brooks, F. P. (1987). No Silver Bullet Essence and Accidents of Software Engineer.

Chen, L. (2017). Continuous Delivery: Overcoming adoption challenges. *El sevier*, 15.

Díaz, E. (2019). Mejora del proceso de desarrollo de una empresa de Servicios de Información mediante la incorporación de Devops. (*Tesis de Maestría*). Universidad de Chile, Santiago De Chile.

Felidré, W., Furtado, L., da Costa, D., Cartaxo, B., & Pinto, G. (2019). Continuous Integration Theater. *IEEE*, 10.

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. New York: 2010.

Kumar AR, A., & TN, S. (06 de 06 de 2020). Automation of Software Development using. *IJERT*, 3.

Makam, V. K. (2020). Continuous Integration on Cloud Versus on Premise: A Review of Integration Tools. *Advances in Computing*, 6.

Pinheiro Luz, W., Pinto, G., & Bonifácio , R. (2018). Building a collaborative culture: a grounded theory of well succeeded devops adoption in practice. *ACM/IEEE*, 10.

Pinto, G., Luz, W., & Bonifacio, R. (2019). Adopting DevOps in the Real World: A Theory, a Model, and a Case Study. *Journal of Systems and Software*, 36.

Rahman, A., Agrawal, A., Krishna, R., Sobran, A., & Menzies, T. (2017). Continuous Integration: The Silver Bullet? 11.