INTEGRACIÓN Y DESPLIEGUE CONTINUO CON DEVOPS COMO CULTURA EN EMPRESAS DEL SECTOR TI COLOMBIANAS

Integration and continuous deployment with devops as a culture in colombian IT sector companies

GILBERTH NICK CRUZ GONZÁLEZ[1], JOSÉ ALEJANDRO FRANCO CALDERÓN[2]

Recibido:02 de diciembre de 2022. Aceptado:28 de diciembre de 2022

DOI: http://dx.doi.org/10.21017/rimci.2023.v10.n19.a132

Resumen

La integración y despliegue continuo son prácticas que han permitido que el ciclo de desarrollo de software se comporte de una manera más ágil, esto gracias a que constantemente brindan la posibilidad de automatizar procesos con el fin de mejorar tiempos y calidad. Este tipo de prácticas requieren de conocimiento y en la mayoría de los casos algún proveedor que cuente con herramientas y recursos con los cuales se puedan realizar dichas automatizaciones, es por esta razón que las empresas colombianas cuentan con opciones para todo tipo de objetivos y presupuestos económicos, desde empresas mundialmente conocidas como lo son Amazon Web Services, Google Cloud Platform y Microsoft Azure, pero al mismo tiempo existen herramientas de código abierto con comunidades activas que son bastante conocidas y utilizadas como alternativas a estas grandes empresas como lo es el caso de Tekton, Jenkins, entre otros. Este artículo de revisión tiene como objetivo identificar cuáles son las ventajas, desafíos, herramientas, entre otros aspectos que caracterizan este fenómeno y por qué ha crecido de tal manera que las empresas buscan opciones para implementarlo, al finalizar este artículo se lograron identificar diferentes herramientas y casos de éxito que brindan seguridad a futuras empresas que pretenden continuar con el mismo camino de tecnologías aplicadas con el fin de buscar el cambio teniendo como base nuevas formas de operar un mismo proceso.

Palabras clave: DevOps; DevSecOps; Integración Continua; Despliegue Contino; Ciclo de Vida del Software.

Abstract

Continuous integration and deployment are practices that have allowed the software development cycle to behave in a more agile way, thanks to the fact that they constantly provide the possibility of automating processes in order to improve times and quality. These types of practices require knowledge and in most cases a provider that has tools and resources with which to carry out said automations, it is for this reason that Colombian companies have options for all types of objectives and economic budgets. , from world-renowned companies such as Amazon Web Services, Google Cloud Platform and Microsoft Azure, but at the same time there are open source tools with active communities that are well known and used as alternatives to these large companies, such as Tekton , Jenkins, among others. This review article aims to identify the advantages, challenges, tools, among other aspects that characterize this phenomenon and why it has grown in such a way that companies are looking for options to implement it, at the end of this article it was possible to identify different tools and success stories that provide security to future companies that intend to continue with the same path of applied technologies in order to seek change based on new ways of operating the same process.

Keywords: DevOps; DevSecOps; Continuous Integration; Continuous Deployment; Software Life Cycle.

1 Estudiante de Ingeniería de Sistemas. ORCID: https://orcid.org/0000-0002-8740-7713 Correo electrónico: gm.cruz@urepublicana.edu.co

2 Ingeniero Electrónico y Magister en electrónica de la Escuela Colombiana de Ingeniería "Julio Garavito", especialista en diseño de aplicaciones para televisión digital interactiva y en administración de tecnologías de la información para la comunicación virtual de la Universidad Manuela Beltrán, Docente investigador de la facultad de ingeniería adscrito al Grupo de Investigación, Desarrollo e Innovación Sostenible - GIDIS de la Corporación Universitaria Republicana. ORCID: https://orcid.org/0000-0003-3931-2186 Correo electrónico: alejing@urepublicana.edu.co

I. Introducción

Actualmente debido a eventos de fuerza mayor como la pandemia y la presencia cada vez más fuerte de ecosistemas virtuales, han logrado evidenciar como la tecnología es cada vez más importante para que las empresas se puedan adaptar al cambio, permitiendo que mediante aplicaciones y desarrollos web todo se encuentre al alcance de las personas, esto lleva a la problemática con la cual se están enfrentando las empresas de TI o centradas en el desarrollo de software, el cual es la gestión del ciclo de un desarrollo tradicional, que lo caracteriza una serie de pasos y tiempos excesivamente largos para lograr completar objetivos de desarrollo, es por esta razón que las empresas requieren que se adapten a procesos que se puedan realizar de una forma cada vez más ágil, es aquí donde surgen distintas metodologías y culturas que fomentan mayor colaboración en los equipos de desarrollo con el fin de mejorar sus entregas y calidad de lo que se desarrolla[1][2].

Una de las metodologías más conocidas y preferidas por las empresas en la actualidad es DevOps, debido a la cultura que fomenta en procesos de automatización, calidad, comunicación y seguridad, estas características se encuentran derivadas de casos que han demostrado tener éxito en empresas altamente conocidas en el mercado, las cuales brindan seguridad a nuevas compañías al aplicar esta metodología como parte de su cultura empresarial[2].

El objetivo principal de este artículo es poder identificar como esta cultura y las buenas prácticas ayudan en el desarrollo y crecimiento de empresas del sector TI en Colombia, generando diversas oportunidades y mejorando su propio ecosistema[1].

II. Historia

A. Inicio, ¿por qué y cómo?

Desde los inicios del desarrollo de software se conocen demasiadas metodologías tradicionales las cuales se centran en un entregable a mediano o largo plazo, para llegar a esto se debe considerar un punto A y punto B el cual es el único camino para completar el desarrollo de la aplicación deseada, a esto se le conoce como ciclo del desarrollo de software, esto significa que en el principio del proceso se contaba con guías que se conformaban por una serie de pasos que se complementaban estructuralmente pero que a su vez no generaba la mejor productividad o resultado.

Dentro de estos procesos que identificamos como "tradicionales" podemos encontrar como la seguridad informática junto con el desarrollo de software ha presentado gran avance con el fin de mejorar el ciclo de desarrollo y la calidad de este, pero para entender a qué se debe estos avances o aportes recientes, se debe identificar porque es un tema que sin importar los años no deja de actualizarse.

Se puede relacionar el "por qué" con un caso reciente como lo es la pandemia, ya que el mundo no se encontraba preparado para afrontarla, lo que permitió encontrar procesos de aprendizaje y mejora continua cada vez más rápidos, se deduce entonces que las empresas sin importar el sector en el que trabajen pretenden mejorar los tiempos de respuesta, desarrollo y soporte, permitiendo a su vez tener un mejor manejo de herramientas tecnológicas y garantizar la seguridad de la información que se desea manipular[3].

De las metodologías tradicionales se pueden mencionar las siguientes:

- Modelo en cascada.
- Modelo en espiral.
- Modelo incremental.
- Modelo basado en componentes.

Las metodologías mencionadas anteriormente se consideran tradicionales y las más utilizadas en los comienzos del desarrollo de software. En la actualidad existen empresas conservadoras que prefieren aplicar este tipo de metodologías. Pero a raíz de las limitaciones que estas presentan surgen las conocidas metodologías agiles[2] las cuales impactan por la adaptabilidad que poseen para incorporarse y mejorar los tiempos de manejo de un proyecto de forma significativa, y gracias a esto surge una nueva incógnita y es ¿Cómo lo hacen? Fig. 1.

DevOps combina las áreas de desarrollo y operaciones con el fin de mejorar el proceso que se conoce como desarrollo de software[3], dejando aún lado lo tradicional y empezando con estos desarrollos y entregas agiles que lo caracterizan.

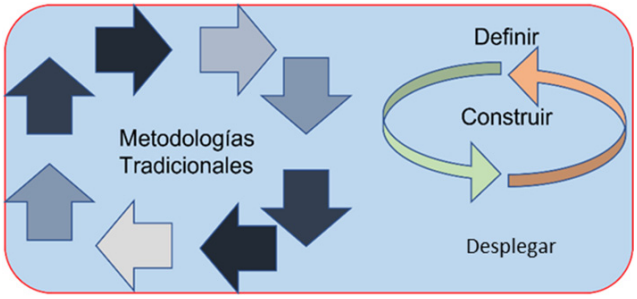

Fig. 1. Ágil vs. Tradicional. Fuente: Elaboración propia.

Todas estas definiciones surgen de la charla brindada por John Allspaw y Paul Hammond en la conferencia O'Reilly Velocity en el 2009[3], es desde este momento que se empieza a conocer el famoso termino DevOps, pero años más adelante con la implementación de este proceso en empresas conocidas, es donde se ve el gran potencial que posee y las posibilidades que brindaría a una empresa, en la actualidad vemos como las empresas prefieren herramientas recientes y que continuamente estén mejorando en desarrollos que quedan estancados por falta de tiempo, es por esto que la práctica y aprendizaje del mundo DevOps cada vez se ve más aplicado en profesionales intentando fortalecer su labor, ya que el aprenderlo no significa que se deje por completo las practicas originales que se tienen, si no por el contrario, permite mantener las esenciales y mejorar las que no poseen el mismo rendimiento que se espera.

B. DevOps como cultura

En la actualidad existen un número significativo de documentos con diferentes palabras intentando conformar una misma definición para este término que es utilizado constantemente. Acerca de DevOps como cultura, en[3] él autor acota que:

A raíz de esta disparidad en las definiciones, surgen diversos estudios que tratan de darle una descripción académica. Según estos estudios, podemos definir DevOps como una cultura que trata de aunar a los equipos de desarrollo y operaciones, basándose en una serie de principios y prácticas que pretenden acelerar las entregas del producto mejorando el feedback de los clientes y la capacidad de reacción ante los cambios.

Además de esto las metodologías tradicionales provocan que las áreas de operaciones y desarrollo se limiten únicamente a cumplir sus labores y metas objetivas mas no enfocarse en las metas globales establecidas, puesto que los métodos de desarrollo tradicionales no son agiles debido a la dispersión de los equipos y la poca colaboración que se presenta entre estos[3].

C. Evolución

Como se evidencia en la Fig. 2 y teniendo en cuenta las estadísticas obtenidas en el reporte realizado por la organización DIGITAL.AI, donde se recopilo información de personas con los siguientes roles: Scrum Master, Project Manager, Development Leadership, Development Team (Architect, Developer, QA, Tester, Ui Designer), External Consultant, Product Owner, Business Analyst, DevOps y organizaciones ubicadas en diferentes ciudades y continentes (Norte América, Sur América, Europa, África, Asia y Oceanía). A continuación se muestra una gráfica representativa de los encuestados, esto con el fin de determinar en qué lugares se ha presentado un aumento de procesos agiles e innovadores.

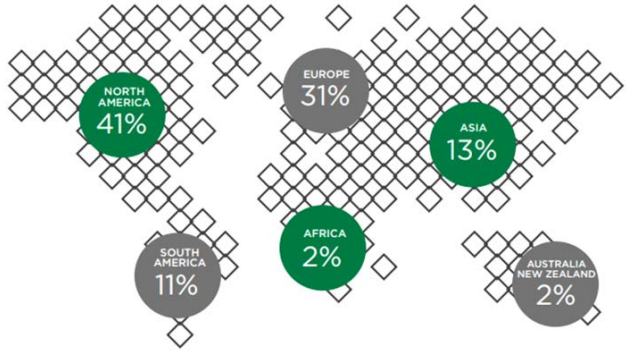

Fig. 2. Adopción de metodologías y prácticas de desarrollo en el mundo[4].

Realizando un análisis del gráfico se puede observar cómo los continentes están empleando procesos ágiles y automáticos en el desarrollo de software, siendo Norte América el más grande respecto a las organizaciones y personas encuestadas con un porcentaje de aprobación del 41%, por otra parte, Oceanía cuenta con el menor porcentaje de aprobación de las organizaciones encuestadas con el 2%, esto en base al estudio realizado a más de 1.120 compañías o personas capacitadas[4].

Se evidencian resultados favorables para las organizaciones que implementaron metodologías y prácticas ágiles como lo son (Scrum, Scrum-ban,

Hybrid Methodologies, Scrum XP/Hybrid, Kanban, Iterative Development, Lean Startup, Extreme Programming, entre otras) indicando de esta manera mejoras en los procesos y el diseño de los sistemas al sustituir las metodologías y prácticas tradicionales, las cuales evidencian un estancamiento en el desarrollo empresarial respecto al mercado actual[5].

En la Fig. 3, se pueden evidenciar las metodologías más implementadas en las organizaciones según el estudio para el desarrollo de software, la metodología líder es Scrum con mayor impacto en las organizaciones del más del 55%.

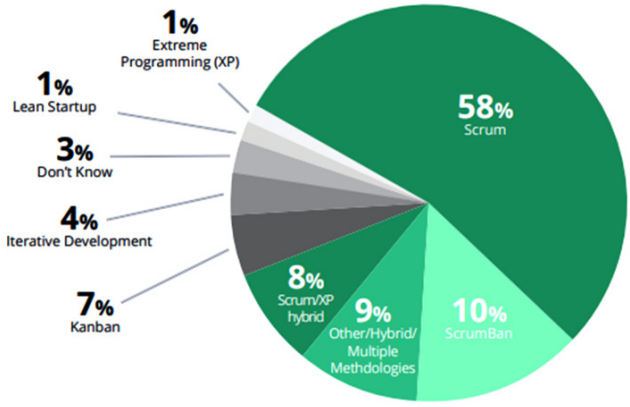

Fig. 3. Porcentaje según las metodologías aplicadas en las organizaciones evaluadas[4].

Algunos de los beneficios de sustituir las metodologías y prácticas tradicionales son:

- Habilidad para gestionar prioridades cambiantes.
- Visibilidad del proyecto.
- Alineación empresa/TI.
- Velocidad de entrega/tiempo de comercialización.
- Mora de equipo.
- Mayor productividad del equipo.
- Reducción del riesgo del proyecto.
- Previsibilidad del proyecto.
- Mejora en la calidad del software.
- Disciplina de ingeniería.
- Gestión de equipos distribuidos.
- Mantenimiento del software.
- Reducción de costos del proyecto.

De las características mencionadas anteriormente se obtuvieron los siguientes porcentajes en organizaciones que implementaron procesos y metodologías ágiles[5][6].

Se puede observar cómo DevOps es conformado en gran parte por procesos agiles que a su vez benefician a diversas empresas al ser aplicados, es por esto que el ciclo de vida del desarrollo de software en DevOps se divide de una manera diferente, pero tiene el mismo fin. Esto permite que los entregables se puedan definir en un menor tiempo y sean constantes, de igual forma se garantiza que al ser un menor tiempo el cambio es una pieza muy pequeña pero que no deja la calidad para completar su fin.

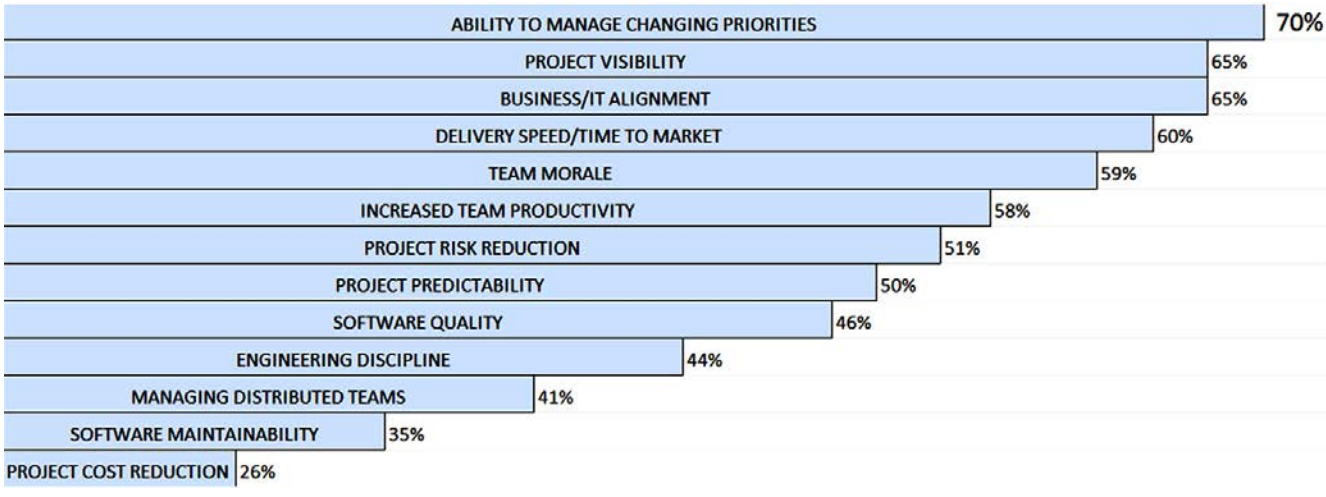

Fig. 4. Beneficios de la implementación de metodologías y procesos ágiles[4].

D. Futuro

Según investigaciones realizadas en los años 2016 y 2017 por Forsgren, Humble y Kim quienes tenían como fin identificar todas las acciones necesarias que una empresa debería aplicar para mejorar su crecimiento, dicha investigación logro evidenciar como el uso de mejores prácticas de software como lo es CI/CD (Continuous Integration/ Continuous Deployment) sin embargo cabe resaltar que en medio de estas prácticas existe un punto medio también identificado como CD este hace referencia a (Continuous Delivery) el cual presenta una gran diferencia con la anterior mencionada (CI/CD), esto admite que estas prácticas y una cultura centralizada en metodologías ágiles permita aumentar significativamente el desempeño de una empresa, mejorando considerablemente la calidad de los procesos de la misma, es por este motivo que constantemente las empresas identifican una guía o serie de pasos que les permita implementar o definir unos tiempos de desarrollo que logre cumplir objetivos en un menor tiempo y con un mayor valor agregado[6][7].

Como se puede observar en la Fig. 5 la elaboración o propuesta de desarrollo de nuevas herramientas que contribuyan a procesos de integración, entrega, despliegue y monitoreos continuos, presuponen una disminución de tiempos, esto gracias al manejo de un mejor soporte que no depende de un proveedor externo.

En la actualidad se han podido evidenciar los cambios en la modalidad de trabajo debido a la pandemia o contingencia a nivel mundial, esto permite que el área de tecnología avance con la creación de herramientas y prácticas para el desarrollo de software los cuales mejoran significativamente los tiempos del ciclo del desarrollo, esto permite que las empresas y el sector en general se prepare para posibles casos que no se han previsto como el COVID 19, evaluando posibles retrasos o retos que estos presenten al intentar adoptar nuevas culturas o prácticas como lo pueden ser[8]:

1. Organización y cultura.
2. Habilidades de aprendizaje.
3. Limitación de recursos.

A pesar de los retos identificados, DevOps y sus prácticas son cada vez más utilizadas por empresas multinacionales, empresas del sector financiero, bancos, empresas del sector tecnológico, entre otras. Para el futuro será una cultura que se volverá prácticamente obligatoria con el fin de que todas las empresas puedan seguir compitiendo en un mismo mercado, y no se queden a un lado debido a lo obsoletas que pueden llegar a ser las metodologías, recursos o herramientas con las que siguen trabajando[9].

Debido a las Fig. 2 y 3 que se han observado en el documento se puede identificar como más em-

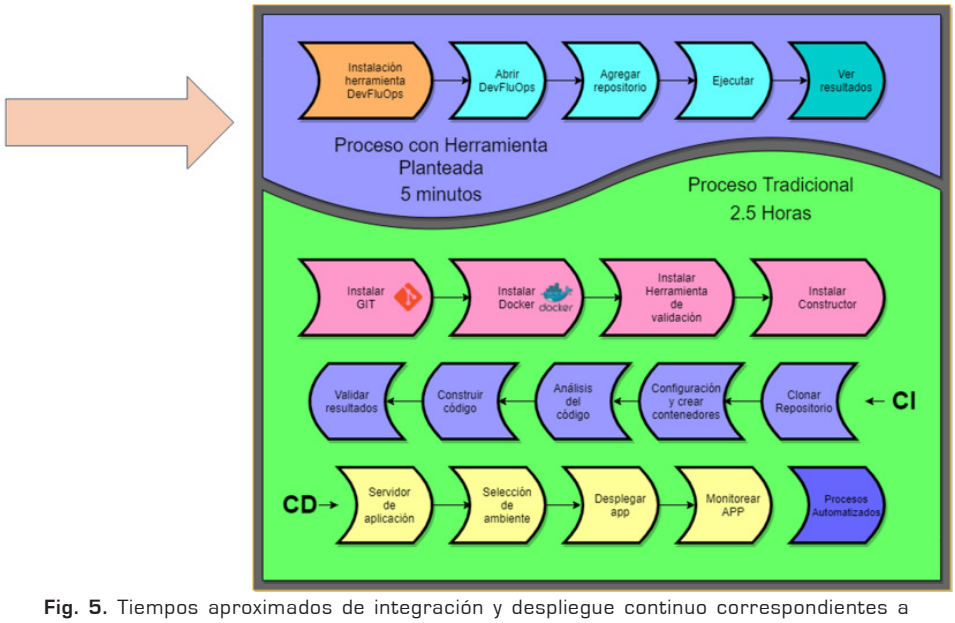

Fig. 5. Tiempos aproximados de integración y despliegue continuo correspondientes a modelos tradicionales y DevOps. Fuente: Elaboración propia

presas se suman al pasar de los años al estudio e implementación de las nuevas prácticas de la mano de metodologías que agilicen procesos, permitiendo que para el año actual (2022) se creen nuevos startups que puedan utilizar las nuevas herramientas con el fin de competir con los grandes del mercado gracias a este tipo de procedimientos.

III. Integración continua

A. Aproximación al concepto

El término "Integración Continua" se originó con el proceso de desarrollo de Programación Extrema (XP) de Kent Beck, como una de sus doce prácticas originales[10]. Es una práctica introducida en XP y luego muy difundida a partir del trabajo de Fow-Ler[5]. Consiste en automatizar y realizar con mayor frecuencia la compilación de los cambios que los desarrolladores hayan realizado en un día, esto a su vez despliega la ejecución de las pruebas unitarias y despliegue de la aplicación ya que todo es un ciclo y dependen unas tareas de otras, esto permite obtener información de retroalimentación inmediatamente después de la ejecución de este proceso[10][11].

La integración continua o también conocida como CI por su siglas en inglés, es un proceso que se utiliza en el desarrollo de software con el fin de que los desarrolladores utilicen una herramienta de control de versiones o repositorio, en el cual como su nombre lo indica permite almacenar y controlar las diferentes versiones con las que puede contar un desarrollo, esto permite que más personas del proyecto tengan acceso a conocer la historia y los avances que se han tenido con cada una de la versiones que se van mejorando a la vez que medir a un determinado equipo de trabajo mediante el sprint el cual es el nombre que va a recibir cada uno de los ciclos o periodos de tiempo que se asigna a un determinado grupo de tareas las cuales son definidas al momento que un proyecto es manejado bajo metodologías agiles como se observa en la Fig. 6 los sprint son definidos con tareas para cada día dependiendo de la duración de este, por lo general al ser metodologías agiles suelen ser de máximo 2 semanas el cual es un tiempo suficiente para realizar cambios suficientes pero no abrumadores a un proyecto, y a su vez poder corregir los errores que se pueden identificar con anterioridad debido a este proceso, la integración continua busca reducir la cantidad de errores sin dejar a un lado la calidad y el tiempo que puede llevar un determinado desarrollo con sus respectivas correcciones[12][13][14][15].

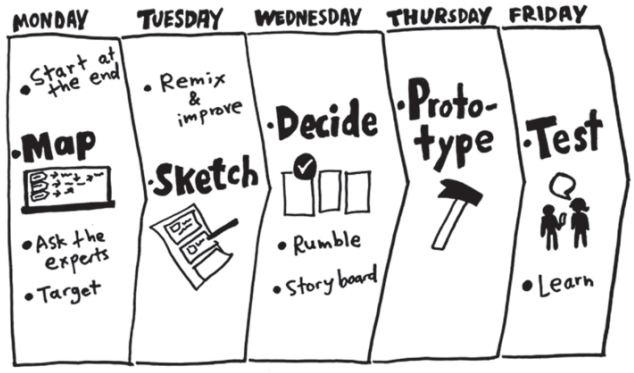

Fig. 6. Sprint[16].

En los desarrollos de hace algunos años era indispensable que los desarrolladores trabajaran y avanzaran lo que más se pudiera para posteriormente realizar una combinación (o merge) en la rama maestra, como se puede observar en la Fig. 7, por lo general sin importar la complejidad del proyecto o número de desarrolladores que se encuentren involucrados se trabaja con varias ramas para posterior realizar un merge (Combinación de cambios en una rama maestra) el cual ubica todos los cambios en una rama master o maestra la cual es la que está atada a un ambiente productivo y donde el usuario final ve los cambios y avances de un determinado proyecto, esto generaba que la identificación de errores no fuera oportuna y estos se acumularan cada vez más reduciendo la calidad del software y aumentando un término conocido como deuda técnica, que lo podemos relacionar al tiempo que se necesita para corregir todo lo anteriormente mencionado[13][15].

Por tal motivo, la integración continúa se volvió un proceso fundamental en el desarrollo de software y es aquí donde se usan diferentes ramas para un mismo fin, cada una definiendo diferentes integraciones según se necesite y permitiendo que cada desarrollador identifique errores antes de que estos lleguen a un ambiente productivo, en la actualidad existen diferentes herramientas que permiten realizar una integración y a su vez tener métricas o reportes que exporten estadísticas de

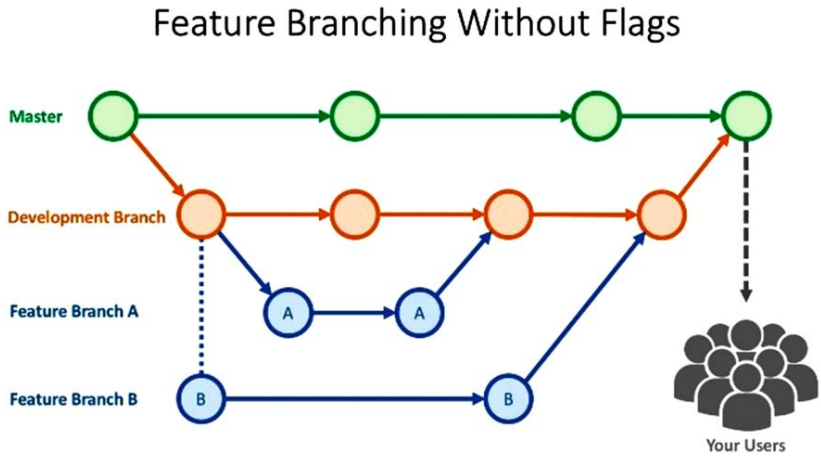

Fig. 7. Ramas[17].

calidad que acceden a tener un crecimiento en el proceso que se realiza, como por ejemplo Azure DevOps una herramienta de Azure enfocada en la nube con métricas constantes, Codecommit una herramienta de AWS enfocada en la nube que cuenta con complementos de otras herramientas que mejora la recolección de eventos o logs, cada una de estas herramientas puede ser complementada por otras externas para mejorar su funcionamiento, hablando de calidad una de las herramientas más utilizadas es sonarqube, el cual analiza el código de forma estática realizando pruebas unitarias y ver que no se esté duplicando o desarrollando con malas prácticas, tiene bastante compatibilidad en la actualidad con casi todos los lenguajes de programación y es fundamental para que los desarrolladores puedan conocer estas métricas de su código[18].

B. Estrategias de Desarrollo

Debido a la iniciativa que han tenido grandes empresas en el mercado como lo son Amazon Web Service de Amazon (AWS), Google Cloud Platform de Google (GCP) y Microsoft Azure de Microsoft, servicios que han permitido que los emprendedores tengan diferentes posibilidades en un mercado tan amplio como lo es el software, dentro de estas estrategias destacan varias herramientas enfocadas a diferentes procesos de la integración continua y según sea el objetivo final permite adaptarse a una mejor que a la otra.

Conociendo las estrategias que se tienen en cuenta para la gestión de proyectos o desarrollo de software como se observa en la Fig. 8, se ha identificado que los proyectos son evaluados o analizados en unas fases más que en otras, las principales en las que se puede apreciar este proceso es en la fase de ejecución la que lleva más tiempo y a su vez costo, de la mano con la calidad e incumplimiento de KPI (Key Performance Indicator) que en español se entiende como Indicador Clave de Desempeño, los cuales son factores claves para cumplir con entregables programados, pero este proceso no solo depende de contar con el dinero y/o las herramientas correctas para ejecutarlo a su vez requiere de un equipo y una organización completa, ya que depende de la madurez de la organización y la correcta ejecución del proceso, es por esta razón que se apoya el uso de una cultura que fomente buenas prácticas como lo son metodologías agiles y reducción de tiempos sin disminuir la calidad del mismo[19][20].

C. Características

La fase principal del ciclo de desarrollo de software basado en metodologías agiles cuenta con características que la identifican como algo fundamental en un proyecto[21]. Adicional a esto, la integración continua se adapta o brinda diferentes características según la herramienta que se escoja, algunas de las principales bajo la filosofía opensource son[22]:

- Github
- Gitlab
- GitKraken

Cada una de estas herramientas cuenta con características como lo son:

- Complementos que se pueden instalar con el fin de ampliar las funciones con las que viene las herramientas.
- Ampliar compatibilidad de lenguajes.
- No tienen costo.
- Personalizable por desarrolladores al ser código abierto.

Al igual que las herramientas de pago cuentan con funcionalidades especiales o únicas como lo son:

- Soporte dependiendo de la suscripción que se pague, en algunos casos 24/7.
- Herramientas y reportes personalizados para garantizar seguimiento al avance del proyecto.
- Backups o formas de recuperar información de una manera sencilla.
- Repositorios con almacenamiento ilimitado.
- Seguridad y notificaciones.

D. Buenas Prácticas

En[16] las buenas prácticas están definidas como un conjunto de pasos o guías que se centran en realizar un mismo proceso con un diferente enfoque, permitiendo que estas generen un valor agregado y calidad a dicho proceso, este tipo de guías son más conocidas como metodologías de trabajo y para un ciclo de vida de desarrollo enfocado o centralizado en DevOps requiere de unas metodologías más agiles que otras, es por este motivo que DevOps trabaja muy de la mano con estas metodologías ya que se encuentran basadas en principios tales como:

- Se valoran más los individuos e interacciones que los procesos y herramientas.
- Se valora más el software en funcionamiento que la documentación excesiva
- Se valora más la colaboración con el cliente que la negociación contractual.
- Se valora más la respuesta ante el cambio al seguimiento de un plan.

Las metodologías de desarrollo tradicionales imponen una disciplina de trabajo fundamentada en la documentación sobre el proceso de desarrollo de software, se realiza un hincapié en la planificación global y total de todo el trabajo a realizar, luego, una vez que esté detallado, comienza el ciclo de desarrollo de software; caso contrario a lo que respecta a las metodologías de desarrollo ágiles, que muchas veces intenta evitar la documentación y se centra en el trabajo, buscando el equilibrio entre proceso y esfuerzo[20].

Las principales características de las metodologías ágiles son:

- Poca documentación.
- Simplicidad.
- Análisis como una actividad constante.
- Diseño evolutivo.
- Integraciones.
- Testeos diarios.

Aunque son más las ventajas que las desventajas, existen casos puntuales en los cuales no es muy favorable utilizar las metodologías ágiles, debido a que no existe una metodología universal que se adapte al alcance o propósito de todos los proyectos en general, a pesar de que las metodologías ágiles ofrecen una solución casi ideal para la gran mayoría de proyectos, existen las conocidas como "tradicionales" que como se puede observar en la Fig. 8 cuentan con más fases pero con un mismo fin, son más conservadoras y a su vez mas detalladas para algunos clientes, todo depende del alcance o necesidad que este tenga, depende de la cantidad de tiempo con el que se cuente para desarrollar o cumplir con un determinado plazo[23].

Fig. 8. Ciclo de metodologías[24].

En la actualidad la mayoría de las empresas utilizan metodologías ágiles debido a sus diferentes características y de la mano de estas surgen cada vez más herramientas que ayudan y mejoran procesos de CI/CD (Continuous Integration/ Continuous Deployment) los cuales le permiten a las empresas generar un mayor valor en sus desarrollos; en la mayoría de casos se hace uso de herramientas de uso libre, ya que cuentan con grandes comunidades que brindan soporte activamente y aun bajo costo para quienes lo operan, claro está, sin dejar de lado la seguridad, dichas herramientas ofrecen un gran servicio por un valor casi nulo.

E. Tecnologías

Como se ha resaltado antes, la tecnología es un área que al pasar de los años evoluciona junto con las industrias adaptándose a los nuevos requerimientos o necesidades que se presenten, por esta razón diferentes empresas y comunidades buscan un continuo crecimiento con el fin de mejorar y a su vez disminuir la cantidad de procesos manuales que se pueden llegar a realizar en un determinado proceso, esto fomenta el uso de nuevas herramientas en su gran mayoría de código abierto para complementar el funcionamiento de herramientas que se pueden quedar cortas según los objetivos de la empresa. Las empresas colombianas que se encuentran involucradas en el mundo de la tecnología cuentan con dos posibles caminos para crear y administra la infraestructura que se encargada de distribuir el software que se desarrolla, los cuales son[25]:

- **On-premise**: Cuenta con instalaciones personales y características prestablecidas de hardware.
- **Cloud**: Cuenta con soporte ilimitado por parte del proveedor que se seleccione y se tiene la posibilidad de escalar según sea el crecimiento de la empresa.

F. Cloud

Según el artículo presentado por Andrés Leonardo Oviedo en[26], este expone las ventajas que tiene el manejo en la nube (o Cloud) estas características resaltan el costo, tiempos e infinita posibilidad de escalamientos vertical u horizontal según se requiera, al escoger una nube con la cual se desea trabajar en la mayoría de ocasiones estas ya cuentan con todo un ecosistema de herramientas y complementos de integración continua que ayudan a completar el ciclo de desarrollo de software que se maneja en las metodologías agiles[27], como se puede observar en la Fig. 8, sin embargo la mayoría de la comunidad de desarrolladores promueve el uso de software libre debido a su fácil acceso y en la mayoría de casos su amplia compatibilidad con nubes existentes, permitiendo migrar de una a otro si así se desea. Las herramientas más utilizadas son:

- Jenkins
- Jenkins x
- GitLab
- BuildKite
- ArgoCD
- Tekton
- Semaphore

Son las más utilizadas debido a su amplia compatibilidad y su soporte por comunidades activas que son conformadas por los mismos desarrolladores y creadores de la herramienta, como se observa en la Fig. 9, la mayoría de herramientas que se nombran cuentan con compatibilidad sin importar el ecosistema que se use, ya sea un ambiente local para pruebas, o un entorno productivo como lo son los cluster de kubernetes que son el nuevo modelo de servidores para desplegar aplicaciones, kubernetes es un orquestador de contenedores y permite que las empresas a través de microservicios desplieguen su aplicación en este servidor teniendo mayor disponibilidad y monitoreo del tema, un contenedor es algo que contiene un software de cualquier aplicación que se desee, a pesar de que estas herramientas son de código abierto la mayoría cuenta con una versión premium en la cual incluyen más funcionalidades que dan cierta exclusividad por un moderado precio en algunas ocasiones.

Fig. 9. CI/CD[28].

G. On – Premise

Un concepto que viene desde los inicios del manejo de creación y despliegue de infraestructura con el fin de alojar un determinado sitio o aplicación, contar con un servidor en las instalaciones propias de la compañía o también conocido como on-premise es un modelo de software el cual consiste en contar con un lugar físico donde se tengan todos los implementos de hardware para desplegar un determinado desarrollo, debido al alto costo de personal e infraestructura que genera es uno de los modelos que ya casi no se aplica para las empresas de este sector[29].

Como se aprecia en la Fig. 10, en esta se observa la complejidad y el costo que puede tener cada una de las posibles infraestructuras con las que se puede aprovisionar una determinada empresa, sin embargo centrados en on-premise, se puede evidencias los altos costos que esto puede llevar, desde instalaciones y hardware, hasta personal capacitado que pueda brindar el soporte y mantenimiento que este requiera, es por este motivo que las nuevas empresas y antiguas en el sector del software prefieren en la actualidad pagar un determinado valor por una mensualidad con la que le garantizan el funcionamiento de la misma manera, pero a un menor costo y soporte garantizado.

Fig. 10. Diferencias entre Cloud y On-premise[29].

IV. Entrega continua

A. Aproximación al Concepto

Según Humble y Farley en[30], CD (Continuos Delivery o Entrega Continua en español) fue creado para resolver el siguiente interrogante: Si alguien tiene una buena idea, ¿cómo se la hace llegar a los usuarios lo más rápido posible?

Es por esta razón que la entrega continua se conoce como una fase extendida de la anteriormente mencionada integración continua, ya que, su objetivo esta alineado con el poder entregar o desplegar cambios no tan grandes a sus clientes[31].

Fig. 11. Entrega Continua. Fuente: Elaboración propia.

Como se observa en la Fig. 11 la entrega continua automatiza todo el proceso realizando pruebas que ayudan a la última compilación o empaquetamiento del desarrollo o también conocido como publicación de software[32]. Cada revisión efectuada activa un proceso automatizado que crea, prueba y almacena la actualización. La decisión definitiva de implementarla en un entorno de producción en vivo la toma el desarrollador[32][33].

Como se ha mencionado anteriormente, entrega continua no es lo mismo que despliegue continuo, es solo una fase evolucionada de la integración continua, lo que no significa que por ejemplo, 20 cambios diarios se deban de ejecutar 20 pasos a producción, por el contrario la entrega continua simplifica la cantidad de tiempo y personal que se requiere para un paso a producción garantizando y manteniendo al día el código que se desea pasar en el momento que el Tech Lead o líder del desarrollo decida publicarlo[32][34].

B. Entrega vs Despliegue

En la actualidad las empresas que pretenden trabajar con metodologías agiles, de cierta manera

en algún punto de esta integración se encuentran con estos términos, ya que al hablar de DevOps normalmente se identifica con las siglas CI/CD, pero al leer la información no es claro si se habla de "Continuous Delivery" o "Continuous Deployment" es por esta razón que depende de la empresa o grupo de personas elegir qué es lo que se desea lograr[31].

La entrega continua está directamente relacionada con estándares de calidad y eficiencia que de por si las metodologías agiles ya manejan, pero que sin importar el desarrollo que se quiera lograr se debe de cumplir con los KPI o ANS (Acuerdos de Nivel de Servicio) los cuales son definidos sin importar la metodología que se aplique[35].

Es por esta razón que las metodologías agiles se enfocan en trabajar en periodos de tiempo más pequeños, con entregables mucho más cortos, permitiendo mejorar la calidad y un buen cumplimiento de un determinado tablero de tareas.

Una de las metodologías más utilizadas es scrum, debió a su fácil adaptabilidad para la mayoría de proyectos, en scrum se resalta por manejar tiempos cortos como lo mencionamos anteriormente, este es conocido como sprint por lo general suelen ser periodos de tiempo cortos ya que los entregables no son tan grandes pero a que su ves estos permiten alcanzar las metas del proyecto en el tiempo planteado, existen diferentes software que ayudan a medir el cumplimiento de estas tareas asignadas, uno de lo más utilizados es JIRA, es un software enfocado en gestión de proyectos mediante tableros en los que se asignan tareas.

En los tableros se pueden encontrar 3 estados a los cuales mover una tarjeta que ha sido aginada a una persona, estos son:

- **To-do**: Son todas las tareas pendientes por realizar, las cuales no han tenido ningún avance por parte de la persona.
- **In-Progress**: Son las tarjetas o tareas que ya se han iniciado, pero no se han finalizado.
- **Done**: Son las tareas que cada uno de los desarrolladores completa y son posteriormente analizadas por las personas encargadas.

Por lo general estos tableros son como se observa en la Fig. 12, este tipo de herramientas permiten tener una mejor organización permitiendo que el desarrollo de un software cumpla con los tiempos establecidos y se puedan medir diferentes aspectos durante el desarrollo.

Debido a que se genera un tablero para cada proyecto es normal encontrar tarjetas que pueden parecen duplicadas, pero que tienen un motivo para estar creadas de esa manera, dependiendo del desarrollo DevOps plantea diseñar diferentes ambientes con el fin de mejorar estándares de calidad y seguridad en cuanto a pruebas funcionales y no funcionales del sistema, esto con el fin de entregar un producto limpio en un ambiente productivo, normalmente en los tableros se crean tarjetas que se diferencian por las siglas correspondientes a cada ambiente, esto permite identificar y saber qué ambiente se trabaja y en qué proceso se encuentra, al finalizar un sprint el tablero debería de estar al 100% y posterior se realizaría un análisis de cómo mejorar para el siguiente sprint.

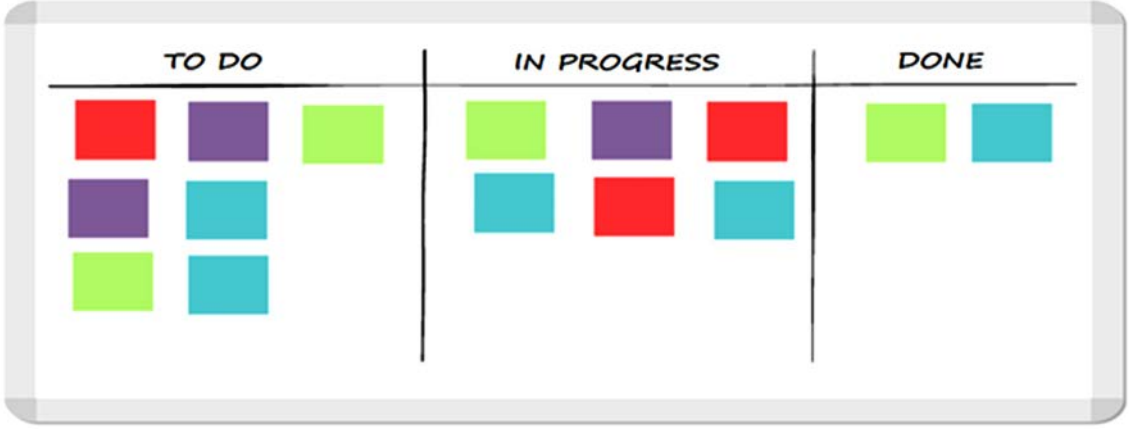

Fig. 12. Manejo de tarjetas[36].

C. Ventajas y Desventajas

En[35] se destacan las principales ventajas, estas son:

- Un menor riesgo en los despliegues, como se ha mencionado con anterioridad, al tener control sobre una serie de cambios pequeños es mucho más sencillo ser analizados antes de ser enviados.
- Seguimiento a los avances y/o cumplimiento del sprint, observar el cumplimiento de tareas asignadas a cada desarrollador es fundamental en un sprint, y la entrega continua permite que a medida que un desarrollador avance el cliente vea cambios y no deba esperar a que se termine todo un desarrollo.
- Mejores productos, permite desarrollar un mejor entregable debido al feedback o comentarios de los cambios, permitiendo reajustar y mejorar continuamente su proceso.
- Mayor calidad, mediante diferentes herramientas o complementos se puede generar un entorno de monitoreo continúo asegurando la disponibilidad y calidad de lo que se desarrolla.

Al igual que todos los procesos o herramientas para el desarrollo de software, también se cuentan con desventajas que varían dependiendo del propósito de cada empresa[32].

- La entrega continua requiere decisiones de despliegue manuales.
- Requiere un personal disponible según los pasos programados a cada ambiente.

V. Despliegue continuo

A. Aproximación al Concepto

Para entender el valor agregado que genera el despliegue continuo en el ciclo de desarrollo de un software se debe de entender y/o conocer cómo se realizaban estos despliegues en un principio[37].

Despliegue tradicional, se debe tener claro que en un enfoque tradicional hablamos de procesos manuales y un equipo completo para llevar a cabo cualquier operación que afecte el funcionamiento de un software en un ambiente productivo, al igual que en la actualidad se puede dividir en construcción del código, pruebas y despliegue de una forma muy general[37].

Cada uno de los 3 procesos en los que se dividió el despliegue tradicional cuenta con subprocesos que se ejecutaban de una forma periódica normalmente con fechas lejanas por los amplios cambios que se aplicaban en un mismo paso.

- Construcción del código: se ejecutaban compilaciones manuales con los respectivos complementos que estos se necesitaran.
- Pruebas: desarrolladores encargados de ejecutar pruebas unitarias, estáticas y funcionales.
- Despliegue: integración del código de los diferentes desarrolladores del equipo, entregas trimestrales o semestrales según acuerdo con el cliente, despliegue en ambiente productivos de forma manual.

En la fase de construcción del código se presentan conflictos debido a las dependencias o librerías que usa cada desarrollador a su conveniencia, reduciendo la eficiencia en los tiempos que se desean.

En las pruebas se debe tener disponibilidad del desarrollador, con tal de probar las funcionalidades que se solicitaron y posterior los respectivos ajustes que se requieran deben ser tenidos en cuenta antes de llegar al día del paso a producción.

En los despliegues, encontramos que no se puede salir a producción hasta que los anteriores pasos se hayan ejecutado y superado los estándares de calidad que se tenían pensados.

Es aquí donde llega el gran cambio para las empresas que aún desarrollan bajo metodologías tradicionales, ya que como se ha mencionado a lo largo del documento DevOps se centra en romper barreras de comunicación entre áreas, trabaja con tiempo de entrega y cambios pequeños, y la principal es la metodología ágil la cual es adaptable y mantiene a un cliente actualizado en la forma que se desea.

Despliegue ágil, teniendo claro los pasos del ciclo de vida de CI/CD se pueden evidencia como la construcción se puede realizar de diferentes

maneras, pero de forma automática, teniendo una imagen base en la cual se trabaja y esto permite que todos cuenten con los mismos complementos o librerías, herramientas de versionamiento de código como los son Git, GitHub, entre otras, permiten centralizar mediante ramas unos estándares o implementos para el desarrollo.

Pasando a las pruebas, en la actualidad existen herramientas open-source o de pago que permiten que mediante un paso posterior a la construcción del código este sea analizado y genere reportes cumpliendo indicadores con el fin de que cada desarrollador analice su propio trabajo, esto permite tener la mayor cobertura posible sobre el proyecto realizando diferentes pruebas al mismo tiempo, sin embargo grandes empresas prefieren tener una persona de testing, la cual se encarga de llevar cada uno de los desarrollos al límite, con el fin de identificar bugs o posibles problemas de compatibilidad que pueda encontrar un usuario final.

Al trabajar con metodologías agiles se facilita el manejo de un proyecto con los desarrolladores, contando con herramientas como Git o GitHub que con repositorios de código abierto permiten almacenar el código completo del proyecto, esto genera mejores tiempos para cada desarrollador y a su vez que puedan avanzar en su proceso sin retrasar a los demás.

De igual manera la entrega continua brinda la posibilidad de que estos cambios siempre estén listos para ser desplegados en un determinado ambiente, esto facilita la visualización y pruebas de las nuevas implementaciones de cada desarrollador a una aplicación[33][37].

Al comparar los dos modelos se pueden evidenciar varios puntos a favor de la metodología ágil, y esto es debido a que la mayoría de clientes buscan un desarrollo lo más rápido posible, ya que no conocen el ciclo que tiene detrás este proceso, como se observa en la Fig. 13 cada una de las fases que conforma el ciclo de CI/CD es característica de contar con una seria de pasos que llegan a un fin, y lo que busca DevOps en conjunto de las metodologías agiles es mantener actualizando al cliente con cambios pequeños pero constantes generando el valor agregado que se espera, es por esta razón que tener claro cuál es el modelo que opera y cuál es el que se desea que opere es fundamental para una empresa teniendo claros estos pasos básicos en el desarrollo de un software[37].

Por la anterior razón surge el despliegue continuo, con el fin de que todo proceso manual sea automatizado para garantizar la disponibilidad del personal para otras posibles labores, este despliegue garantiza que al subir un determinado cambio en cuestión de minutos se pueda ver reflejado en alguna de los ambientes, permitiendo realizar las pruebas con el cliente y que se evalúen los nuevos cambios integrados.

B. Herramientas

En la actualidad existen diferentes herramientas para el despliegue continuo, algunas con más funcionalidades que otras, pero con el mismo fin, las más conocidas provienen de código abierto y nubes conocidas en todas las industrias como lo son Google, AWS y Azure. Dentro de las herramientas más utilizadas se destacan.

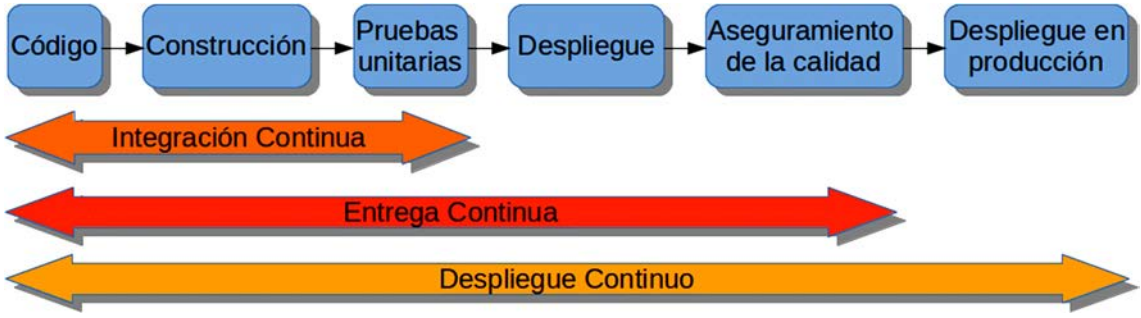

Fig. 13. Despliegue Continuo[38].

- Azure DevOps
- CodeDeploy
- Jenkins
- Tekton
- Github Actions

Como se observa en la Fig. 15 son los símbolos o logos con los cuales se identifican la mayoría de estas herramientas pero que sin importar cual se escoja todas responden con un correcto funcionamiento hacia un sistema en la nube, cada una de las herramientas mencionadas tienen características que las diferencian, pero trabajan con un término similar un "Pipeline" o "Tubería" en español, se toma como un posible camino para llegar a un fin, cada nube o aplicación de CD utilizar el mismo archivo, aunque con diferentes nombres[39].

Por lo general los archivos pipeline son escritos o definidos en un lenguaje conocido como YAML o TOML, en la Fig. 14 se puede apreciar un modelo de pipeline basado en los requerimientos de Azure para su correcto funcionamiento en la plataforma Azure DevOps, este archivo se configura para definir la serie de pasos que debe de seguir un despliegue continuo o lo que se conoce como el ciclo CI/CD, esto se configura la primera vez que se debe desplegar parametrizando las variables y respectivas ramas con las que se va a trabajar las cuales corresponden a ambientes específicos, esta configuración permite que un determinado cliente observe los cambios constantemente que se van añadiendo al desarrollo[39][40].

Por lo general para elegir una plataforma o herramienta de CI/CD se busca un ecosistema que permita automatizar gran parte de tareas, cada una de las herramientas que se consideran más utilizadas cuentan y destacan por contar con funciones que cumplan con el ciclo de desarrollo de software, se pueda trabajar con metodologías agiles y en general con DevOps[41].

Fomentar el uso de herramientas de este tipo se realiza con el objetivo de reducir tiempos de despliegue de software a un ambiente, ayudar a desarrolladores con procesos manuales aprovechando el mejor uso del tiempo disponible y a su vez familiarizar a empresas o a sus mismo clientes de los beneficios de trabajar con metodologías agiles y con DevOps en general[41].

Debido a que las tecnologías, lenguajes y herramientas en el sector de la tecnología cambian constantemente, se requiere de herramientas y metodologías que evolucionen al mismo ritmo, es por esta razón que sin importar si la herramienta es de pago u open source cuente con una comunidad activa que actualiza constantemente su software para permanecer en un buen lugar frente a su competencia.

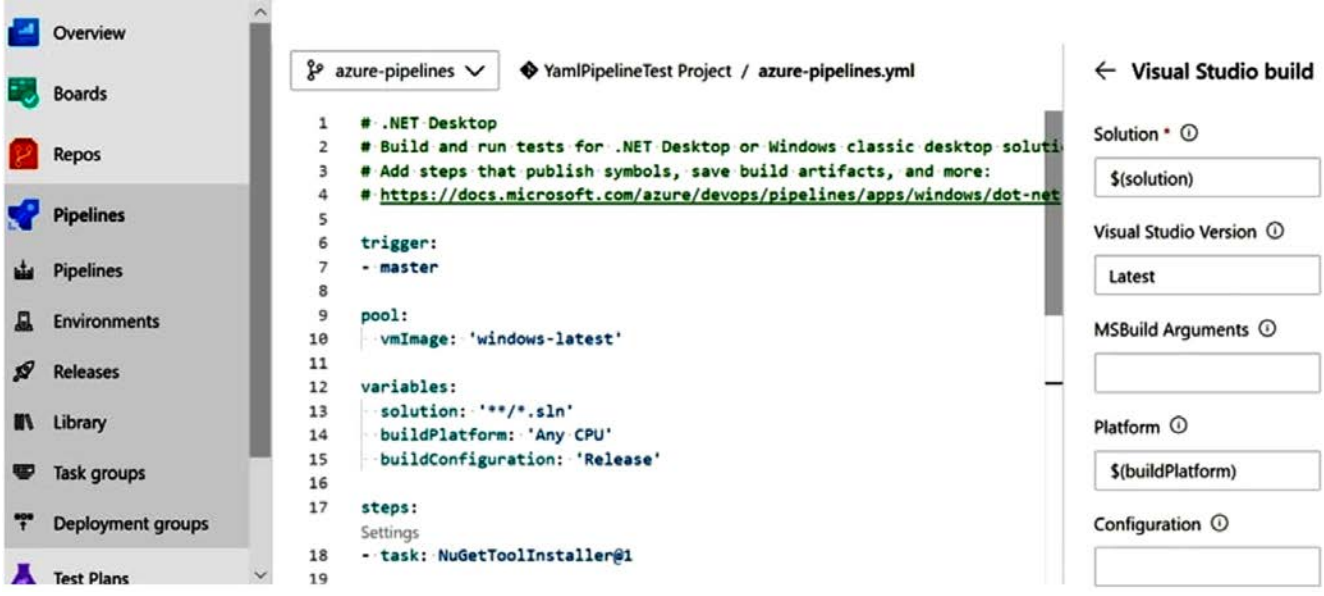

Fig. 14. Pipeline[40].

Fig. 15. CI/CD. Fuente: Elaboración propia.

C. Características

El despliegue continuo o como es conocido por sus siglas en inglés como Continuous Deployment (CD), es la fase final que conforma el ciclo de desarrollo de software para DevOps y las metodologías, en este último paso se busca que mediante los avances de los desarrolladores, los cambios se puedan ver y probar en un ambiente de una manera casi que simultánea, por lo que al llegar a esta fase se configura un pipeline de tal manera que al detectar cambios en una rama de un proyecto, el ecosistema de CI/CD realice un despliegue automático con el fin de enviar los últimos cambios a un ambiente donde los puedan revisar.

A diferencia de la entrega continua, en esta fase si se busca que los cambios se desplieguen de forma automática, por lo general son para ambientes no productivos como desarrollo, donde no se requiere una interacción humana para aceptar dichos cambios, esta fase permite que mediante distintas configuraciones se ejecute una tarea la cual se encarga de recompilar el proyecto, analizar los posibles cambios que se hayan ejecutado y enviarlos directamente al DNS o link público de internet en el que se pueda y se encuentre configurado el ambiente.

En general se puede decir que el despliegue continuo no es más que el resultado de aplicar y configurar de forma correcta la integración y la entrega continua, a pesar de que el despliegue continuo se aplica en ambientes no productivos, no impide que alguna empresa o persona desee configurarlo en producción por automatizar ese proceso.

Esta decisión o implementación deben de ser medidas frente a los beneficios o contradicciones que esta podría traer, sin embargo, debido a las herramientas y posibilidades que ofrece un ecosistema cloud, prácticamente se puede dejar todo en manos de esta configuración y el cumplimiento del ciclo CI/CD en un proyecto, mejorando los tiempos de despliegue, la visualización de estos cambios por parte del cliente y la posibilidad de enfocar al equipo de desarrollo en otros aspectos mientras que el despliegue es ejecutado de forma automática[3][42].

Por otro lado, esta fase o resultado del ciclo cuenta con características o serie de pasos que se deben de ejecutar para su completa realización, estos son[26]:

Fig. 16. Despliegue Continuo[42].

- Pruebas.
- Monitoreo.
- Cambios de reversa.

Como se evidencia en la Fig. 16 las fases que completan este resultado a su vez traen consigo unos beneficios los cuales agregan valor a la empresa y también a sus clientes, ya que garantizan que en cualquier momento se puedan ver los cambios solo con realizarlos, las ventajas más grandes de concluir con despliegue continuo son[16]:

- Satisfacción al cliente.
- Sin grandes errores.
- Eficiencia laboral.
- Menor carga de trabajo.

VI. Impacto en la industria

A. Beneficios

Como se ha podido observar al momento de referenciar algunos artículos científicos que se han consultado, estos presentan impactos que pueden tener los atributos de DevOps en la industria, según algunos de ellos "DevOps en el desarrollo SaaS desde el punto de vista de los expertos" se deben tener en cuenta diferentes atributos como se pueden ver en la Fig. 17[9].

Como se observa en la Fig. 17, se puede concluir bastante información como por ejemplo el impacto que puede tener un atributo vinculado al ciclo DevOps y al ciclo de desarrollo de software ya sea IaaS (Infraestructura como Servicio) o SaaS (Software como Servicio), claramente se puede observar que al incluir este tipo de información se puede tener una vista más clara del objetivo del desarrollo.

Luego de analizar revistas científicas y estados del arte, se puede analizar como con el pasar de los tiempos se intenta definir un modelo o guía el cual facilite la implementación y adopción de DevOps, esto con el fin de identificar los beneficios que puede traer consigo seguir dicha guía, gracias a dichos estudios realizados se han identificado los siguientes beneficios[18]:

1. Productividad.
2. Competitividad.
3. Satisfacción al cliente.
4. Calidad.
5. Monitoreo.

Debido a estos beneficios DevOps es el centro de una conversación de interés en la cual la industria de software es su principal actor, en este tipo de situación es donde las empresas que llevan muchos años en un mercado de software deben de tomar decisiones fundamentales que los pueden beneficiar en el crecimiento y expansión a un mayor rango de la empresa ya que los startups no dudan al aplicarlo porque no tienen mucho que perder y sí mucho por aprender y ganar.

Atributos de DevOps	Sin Impacto	Impacto Bajo	Impacto Medio	Impacto Alto	Impacto Muy Alto
Planificación continua	3%	0%	37%	40%	20%
Revisión automática de Código	3%	3%	37%	40%	17%
Pruebas automatizadas	3%	3%	27%	37%	30%
Prueba de métricas y políticas	3%	0%	27%	47%	23%
Patrón de ramificación	3%	3%	43%	27%	23%
Lanzamientos continuos	3%	3%	20%	43%	30%
Despliegue automatizado	7%	10%	20%	23%	40%
Infraestructura como Código	3%	10%	40%	27%	20%
Retroalimentación	3%	0%	23%	30%	43%
Herramientas automatizadas para monitorea	7%	7%	30%	23%	33%

Fig. 17. Atributos DevOps[9].

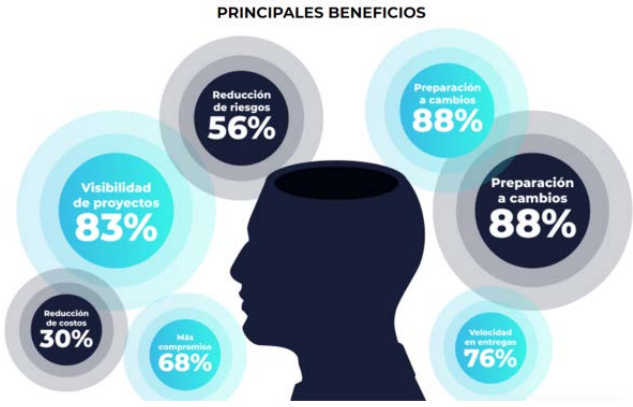

Fig. 18. Beneficios en la industria[43].

Según la revista universitaria "DevOps in Industry 4.0: A Systematic Mapping" estos beneficios se aplican a la industria 4.0 permitiendo reducir las prácticas tradicionales en las fábricas y empresas de software[43].

B. Nuevas Tecnologías

Con el pasar de los tiempos se observa como las diferentes áreas evolucionan en procedimientos, herramientas, alianzas entre otro tipo de procesos que intervienen en del desarrollo de una empresa, teniendo claro este contexto, DevOps no es la excepción y constantemente personas que están en este mundo están desarrollando y mejorando la forma de trabajar mejorando cada vez de una mejor forma el ciclo de vida de desarrollo de software.

Debido a acontecimientos que se han presenciado en los últimos años se ha podido observar como la transformación digital ha cogido más peso en cada una de las áreas que en la actualidad presentan un gran impacto en la industria, gracias a los acontecimientos como la pandemia o migración laboral, permite que las empresas le enfoquen la prioridad a esta transformación para continuar en el mercado[44].

Debido al nuevo tipo de aplicaciones, estas permiten que DevOps este mejorando constantemente procesos de automatización de tareas los cuales en los inicios de un desarrollo de un nuevo producto con tal de mejorar el rendimiento. Se evidencia y ejecutan tareas de forma automática, gracias a estas nuevas tecnologías cada vez más tareas se pueden automatizar disminuyendo la carga de los desarrolladores.

A pesar de que este tipo de culturas no se trata de la cantidad de herramientas que se emplea, en la mayoría de los casos son necesarias para poder garantizar cada uno de los pasos del ciclo de vida del desarrollo como lo son la entrega, la integración o el monitoreo continuo[46], en la actualidad se pueden observar constantemente el uso de herramientas como lo son:

- **Jenkins**: la cual es una de las herramientas más utilizadas por su amplia compatibilidad y gran soporte, al ser una herramienta open source tiene una comunidad bastante activa debido a casos en los que llega a pro-

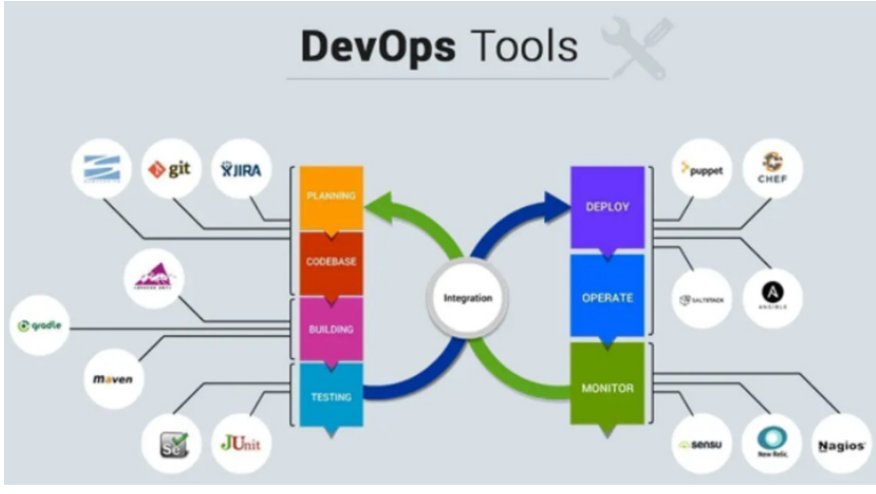

Fig. 19. DevOps Tools[45].

ducción una implementación con este tipo de herramientas.

- **Circle-Ci**: una de las herramientas poco escuchadas, pero muy utilizadas por su solución basada en la nube, multiplataforma y con muchas opciones de personalizar un mismo ambiente de trabajo.

- **Travis**: Al igual que los anteriores, es una herramienta de CI para proyectos open source y cuenta con versión de pago o premium según el tipo de proyecto que se desee manejar.

- **Maven**: Es una herramienta principalmente enfocada en proyectos java.

Como lo se ha visto en artículos científicos anteriormente mencionados, la transformación digital trae consigo demasiadas posibilidades para completar un mismo fin, en este caso hablamos de que cada etapa del ciclo DevOps cuenta con infinitas herramientas tanto open source como de pago que permiten y ofrecen diferentes características que destacan sobre las otras, pero es aquí donde los términos que se han mencionado en un inicio definen el propósito y herramientas posibles que se puedan ejecutar, siempre dependerá del alcance y propósito de desarrollo pensando en un futuro no muy lejano donde se quiere estar con ese software, esto permite que se evalúen diferentes procesos que le permiten a una empresa elegir herramientas correctas.

C. Casos de éxito de la implementación de DevOps como cultura

Como se ha presentado en el contenido del documento, DevOps más que una herramienta es una manera de operar, una cultura que de la mano de buenas prácticas y metodologías agiles busca agregar seguridad, disminuir tiempo y costos, pero sin dejar de lado la calidad.

Por el motivo anterior, se puede tomar como caso de éxito grandes empresas que optaron por el uso de DevOps. El primer caso de éxito que se puede analizar es el de Netflix, una plataforma de streaming mundial la cual tiene una presencia bastante grande en países como una de las plataformas más utilizadas en todo un año, como bien se sabe, esta plataforma se encarga de poner a disposición miles y miles de títulos como películas, series, novelas, entre otros, y es aquí donde alguien se puede preguntar ¿Cómo es posible mantener a tantos usuarios conectados, al mismo tiempo durante todo un año?

Fig. 20. Netflix y DevOps[47].

En 2012 Netflix encontró que manejar estos servicios eran muy costosos de mantener, pero más que eso era prácticamente imposible testear el funcionamiento de todo el ecosistema de desarrollo, es por esto por lo que la compañía requería de un personal que realizan pruebas de forma manual durante un periodo de tiempo constante y considerable con el fin de encontrar vulnerabilidades o errores que se pudieran presentar Este proceso manual traía consigo contradicciones tales como un sinfín de errores que eran identificados pero que a su vez los desarrolladores no se ofrecían para resolverlos debido al tiempo y la complejidad que estos presentaban[47].

Es por esta razón que Netflix paso por varios procesos antes de llegar al lugar en el cual se encuentra el día de hoy, como lo son:

- Conocer el ciclo de vida interno de DevOps.
- Escalar usando herramientas para desarrolladores.
- Buscar personal con conocimiento en cada una de las partes del desarrollo de software.
- Conocer y aplicar las ventajas y minimizar las desventajas de aplicar DevOps.

Una vez evalúalos los ítems anteriores se pudieron empezar a ver los primeros cambios en la estructura a la cual estaban acostumbrados, ya que decidieron que los mismos desarrolladores serían los encargados de operarla, permitiendo identifi-

car y solucionar de forma pronta los errores que se pudieran presentar, Al ver que las velocidades de desarrollar una nueva funcionalidad mejoraron significativamente, vieron como cada proceso mejoro la comunicación entre las diferentes áreas permitiendo entregas mucho más rápidas y de mucha mejor calidad. Con el pasar de los años y las modificaciones al proceso cotidiano o tradicional con el que venía esta compañía, en el Netflix para el año 2018 logro reducir tiempos y costos de procesos en los que se demoraban días, a tal punto de realizarlos en horas y aprovechar de una mejor manera el personal con el que contaba, permitiendo que el sistema tuviera la posibilidad de escalar según sus servicios. Hoy en día a pesar de la competencia que hay en el mercado, Netflix sigue siendo una de las plataformas de streaming más consumidas por usuarios en toda aparte del mundo[47].

Otro caso de éxito de DevOps en la industria es una de las empresas que es pionera en el mundo de la automatización y adopción del ciclo DevOps, esta empresa es conocida por el nombre de Red Hat, y con el pasar de los años esta ha podido replicar su experiencia, conocimiento y cultura a diferentes clientes empresariales que buscan sus servicios, en este caso no haremos énfasis en el éxito que ha tenido Red Hat en el mercado si no puntualmente el caso de éxito de uno de sus clientes, hablaremos de una empresa conocida a nivel mundial por el nombre de Barclays, una empresa de servicios financieros con alcance mundial con sede central en Londres.

Fig. 21. Barclays y Red Hat[48].

Con el objetivo de aumentar la innovación y la productividad, Barclays se puso en marcha para crear una plataforma como servicio para aplicaciones (aPaaS) como parte de su programa de nube. Utilizó Red Hat OpenShift Container Platform y otras soluciones Red Hat para actualizar su infraestructura de TI y adoptar un enfoque ágil de DevOps en el desarrollo de aplicaciones. Barclays adoptó un enfoque DevOps en el cual los equipos técnicos y comerciales trabajan juntos para satisfacer rápidamente las demandas de los clientes y del mercado a través del desarrollo continuo. Con el nuevo enfoque DevOps y el entorno aPaaS, los desarrolladores de Barclays pueden trabajar más eficientemente para ofrecer actualizaciones y funciones de forma rápida. Las capacidades de autoservicio han reducido los tiempos de aprovisionamiento de semanas a horas y han permitido que el personal de TI trabaje en proyectos novedosos y valiosos, en lugar de ocuparse de las tareas rutinarias[48].

De esta manera al igual que proveedores de servicios como Red Hat, en la actualidad diferentes nubes como Azure o AWS brindan esta posibilidad a nuevas empresas que quieran comenzar de una vez con una cultura seria orientada a los procesos de la mano de la automatización, la seguridad y la calidad.

Sin embargo, este tipo de conocimientos y culturas no solo se pueden aplicar en empresas reconocidas a nivel mundial o con un grandes ingresos producto de sus servicios ofrecidos a distintos clientes, este tipo de culturas es adoptadas por empresas emergentes que quieren iniciar en el mundo de TI y sus buenas prácticas, a continuación, veremos algunos casos de éxito de empresas colombianas y otras que a pesar de no ser creadas en Colombia cuentan con sus sedes en el país y aplican este tipo de cultura.

Fig. 22. DoubleVPartners[49].

Como primer caso de éxito a nivel nacional encontramos a la empresa Double v Partners, la cual es una startup del sector de TI que no lleva más de 2 años en el mercado, pero que quiso arriesgarse e iniciar en el campo de DevOps para aprovechar sus ventajas en el mundo de la tecnología, esto permitió que se pudieran involucrar con toda una infraestructura con el proveedor de servicios AWS, y a la fecha pueden administrar y operar su negocio de la mano de servicios complementarios como lo son: Kubernetes o Docker, brindando un mayor

valor a los clientes que contratan esta empresa para el manejo de sus desarrollos e infraestructura. Double v Partners en la actualidad ofrece un catálogo basto de servicios, pero con los que se dieron a conocer y que mejoran cada vez el valor de estos son sus consultoría para TI, desarrollo web, de aplicaciones y manejo de infraestructura para distintos clientes, el modelo que maneja esta empresa se centra en fomentar a sus empleados el modelo que plantea DevOps como cultura, el cual pretende romper las barreras invisibles que existen entre áreas con el fin de que de cierta manera todas las áreas de una misma empresa tengan el mismo conocimiento de lo que hacen o trabajan en otra área[49].

Conociendo la historia de esta empresa, se mencionará algunos de los clientes que han manejado como casos de éxito debido a la adopción de estas metodologías y procesos.

Fig. 23. Movet[50].

Movet, es una veterinaria la cual se le propuso la idea de integrar todos los conocimientos en veterinaria junto con el gran avance de la tecnología llegando al punto en el que están hoy, no sin antes pasar por todo un proceso de transformación en el cual se mejoró la cultura y la comunicación entre cliente y el área de TI mejorando los tiempos de desarrollo de pruebas y demás, permitiendo que a la fecha sigan creciendo significativamente a tal punto de contar con una página web y una aplicación para Android y iOS, este gran crecimiento y continua ganancia para el cliente ha permitido que se posicione en un mejor lugar de tal manera que pudo adquirir sedes físicas con el fin de complementar los servicios con los que ya contaba de manera virtual, DevOps permitió que infraestructura en una nube y de la mano de kubernetes le brindara la posibilidad de seguir creciendo conforme la cantidad de clientes siga aumentando, pero a la vez, sin dejar a un lado la seguridad del sistema permitiendo transacciones seguras como lo son pagos, compras, manejos de inventarios, entre otros, esto debido a metodología de trabajo scrum las cuales fueron aplicadas para el desarrollo de todo lo que existe en Movet en relación a la infraestructura de TI hasta día de hoy[50].

Fig. 24. Bienco[51]

Otro caso de éxito, se puede mencionar una empresas del sector inmobiliario, en este caso se hará mención de Bienco, la cual aprovecho un momentos difícil para el mundo como lo fue la pandemia, mejorando sus desarrollos e infraestructura permitiendo que las personas que pasaban por una situación difícil pudieran buscar su próximo lugar sin tener que salir de la comodidad de su hogar, en este caso DevOps le permitió crear y manejar infraestructura escalable y desarrollos en base a microservicios girando en torno a kubernetes el cual es el encargado de la administración de estos servicios, en la actualidad Bienco es una inmobiliaria enfocada en procesos seguros y de calidad contando con diferentes ambientes para desarrolladores y personal de pruebas, con el fin de mejorar para sus clientes finales en toda la cadena de valor de sus servicios[49].

Fig. 25. Stoyco[51]

Volviendo al sector TI encontramos una empresa nueva la cual se encuentra enfocada en temas relacionados a criptomonedas y Non-Fungible Tokens más conocido como NFT, esta empresa se encuentra centralizada en la compra y venta de

NFT que es conocido como un activo que no puede ser modificado como lo son imágenes, obras de arte, entre otros. Está empresa se une al mundo de la tecnología y el área de activos no tangibles de la mano de DevOps el cual interviene con el fin de aplicar las prácticas y generar una infraestructura escalable y automatizada, ya que al ser un mercado tan grande y con demasiado ingreso en simultaneo de diferentes personas requiere que la infraestructura que se manaje cuente con posibilidad de escalar según lo requiera. Es por esta razón que stoyco opta por implementar todo en un proveedor cloud como AWS y aplicar servicios como kuberntes y herramientas de integración continua como lo es tekton con el fin de potencias y agilizar los posibles desarrollos o mejoras que requieran hacer en su aplicación, página o recursos que tengan desplegado dentro de su infraestructura, al ser una empresa nueva es poco conocida en el mercado, sin embargo esta se posiciona cada vez más dándose a conocer en diferentes eventos en varios países con el fin de mostrar su objetivo y potencial futuro.

Fig. 25. Colpatria[52].

Como se mencionó a lo largo del documento, DevOps no funciona solo en empresas del sector TI o empresas emergentes que quieran iniciar de una vez con una cultura nueva, en Colombia encontramos diferentes bancos que día a día mejoran sus procesos de DevOps con el fin de brindar una mejor experiencia para los clientes y distintos usuarios finales, mencionaremos dos que tienen un gran impacto en el sector financiero en Colombia. El banco Scotiabank Colpatria es uno de los bancos más conocidos por sus diferentes financiamientos en procesos de vivienda, estructuras y presencia en diferentes países del mundo, pero para lograr ser tan renombrado el banco utiliza DevOps como cultura central del área de TI, se caracteriza por mantenerse en constante mejora, esto lo puede observar con sus diferentes aplicaciones y páginas que manejan, a la vez que no dejan de lado la seguridad permitiendo que millones de personas realicen transacciones diarias. Esto el banco lo pudo lograr implementando tecnologías como kubernetes y manejo de herramientas de CI/CD en entornos on-premise con el fin de no desperdiciar la infraestructura con la que ya se contaba, de la misma manera en base a la cultura que fomenta una práctica como DevOps se generaron mejores manejos de tiempos de sprint los cuales permitieron mejorar los tiempos de respuesta a los clientes.

Al hablar de sectores financieros lo primero en garantizar seria la seguridad y la gestión de procesos estrictos y rigurosos, de esta manera, se hace especial énfasis en la calidad de los procesos que interactúan con el usuario final, cuentan con diferentes ofertas que permiten integrarse al equipo de DevOps el cual ha generado un gran cambio en este banco mejorando rendimiento y disponibilidad de páginas y aplicaciones, teniendo pasos a producción controlados, comunicación entre diferentes áreas, entre otras cosas a su favor.

Permitir relacionarse con nuevas herramientas de tecnología y mejorar la disponibilidad de sus sistemas permitió que el banco creciera en solicitudes de tarjetas de crédito, mejoro la comunicación de sus clientes permitiendo que todo proceso lo realizaran 100% de forma digital de la mano de un proveedor de servicios en nube como Google[52].

Fig. 25. Davivienda[53].

Otro caso de éxito en el sector financiero lo presenta el Banco Davivienda, este con presencia en 6 países distintos. Para el caso de Colombia adopta la cultura DevOps de la mano de Google como proveedor de servicios. La transformación digital que adopto el banco Davivienda le dio un gran impulso en ventas y en cantidad de clientes potenciales, permitiendo mejorar las labores diarias de sus trabajadores y la relación de sus clientes con sus ecosistemas digitales. Esta innovación marca desde manejo de infraestructura en nube hasta la reducción de implementos físicos como lo es el papel, cuidando de esta manera el medio ambiente y los recursos que provee.

Davivienda utiliza como proveedor a Google con el fin de garantizar la alta disponibilidad de sus servicios, aplicando a su vez buenas prácticas y utilizando software libre fomentando el uso de este tipo de software y a su vez sacando las ventajas que estos brindan, como son despliegues de

forma continua con Jenkins y pasos a producción controlados en horarios no transaccionales con el fin de siempre ofrecer un servicio disponible, al igual que el anterior banco utiliza cluster de kubernetes administrados por el proveedor de cloud mencionado, separando por ambiente y permitiendo testear y mejorar antes de realizar un paso a producción[53].

En conclusión, se puede observar cómo el sector TI puede sacar un gran provecho mejorando y administrados procesos agiles e infraestructura en nube con el fin de reducir el consumo de recursos físicos, pero esta cultura y procesos no son impedimento para que cualquier empresa quiera aplicarlo, como se observó en el sector financiero e inmobiliario, este tipo de culturas y herramientas permiten que las empresas puedan cumplir sus objetivos propuestos desde un inicio independientemente del sector.

Referencias

[1] R. E. Giachetti, & D. L. Van Bossuyt, Challenges of Adopting DevOps for the Combat Systems Development Environment. Defense AR Journal, 29(1), 22-49. 2022.

[2] Y. Franco Noreña, Revisión sistemática de la literatura sobre tecnologías Docker.

[3] J. J. Padrón Hernández, DevSecOps: integración de la seguridad en entornos CI/CD.

[4] Digital.ai, «15th Annual State Of Agile Report».[Internet].2021.

[5] V. Celi & O. Stalin, Entorno de integración, entrega y despliegue contínuo de software en la Universidad Católica Sede Esmeraldas (Doctoral dissertation, Ecuador-PUCESE-Escuela de Sistemas y Computación). 2021.

[6] K. López, Orquestación de herramientas de integración continua en lenguajes funcionales. 2019.

[7] C. B. Pareja Valerio, & L. J. Burgos Robles, La arquitectura de software basada en microservicios: Una revisión sistemática de la literatura. 2019.

[8] J. Castillo, A. Martínez, C. Quesada-López & M. Jenkins, Caracterización de las prácticas de DevOps en organizaciones que desarrollan software: Un mapeo sistemático de literatura. Revista Ibérica de Sistemas e Tecnologias de Informação, (E28), 83-96. 2020.

[9] R. Prosper-Heredia & M. Vargas-Lombardo, DevOps en el desarrollo SaaS desde el punto de vista de los expertos. Revista Ibérica de Sistemas e Tecnologias de Informação, (E34), 252-263. 2020.

[10] M. Fowler. (Continuous Integration[Online]. Disponible: http://martinfowler.com/articles/continuousIntegration.html. 2006.

[11] M. Mascazzini & G. Dapozo, Desarrollo de una aplicación web utilizando Desarrollo Guiado por Comportamiento e Integración Continua. In XVIII Concurso de Trabajos Estudiantiles (EST 2015)-JAIIO 44 (Rosario, 2015). 2015.

[12] I. P. Chávez Martínez, Uso de herramientas de integración continua para reducir el tiempo de despliegue de procesos de Big Data en una entidad financiera. 2021.

[13] AWS, A. W. ¿Qué es la integración continua? 2021. Disponible en: https://aws.amazon.com/es/DevOps/continuous-integration/

[14] A. Requena Mesa, Que es un Sprint. 2018. disponible en https://openwebinars.net/blog/que-es-un-sprint-scrum/

[15] DevOpslatam. "Fabrica de Software"[internet]. 2022.

[16] Tipos de metodologías ágiles de aplicación empresarial: Design Sprint. Disponible en https://franciscotorreblanca.es/metodologias-agiles-design-sprint/

[17] ¿Qué es Git? Obtenido de http://webipedia.es/tecnologia/cursos/git-tutorial-que-es-github/

[18] J. Guerrero, C. Certuche, K. Zúñiga & C. Pardo, What is there about DevOps? Preliminary Findings from a Systematic Mapping Study. 2019.

[19] S. A. Tamayo Heredia, R. Machado Pedraza, R. Delgado Victore, & J. Menéndez Rizo. El sistema de Integración Continua en la Gestión de Proyectos. 2018.

[20] P. Cáceres, E. Marcos & G. Kybele, Procesos ágiles para el desarrollo de aplicaciones Web. Taller de Web Engineering de Las Jornadas de Ingeniería Del Software Y Bases de Datos. 2001. http://dlsi.ua.es/webe01/articulos/s112.pdf.

[21] Tibco. ¿Qué es despliegue continuo?. 2022. https://www.tibco.com/es/reference-center/what-is-continuous-deployment

[22] A. Slaughter, J. W. Peterson, D. R. Gaston, C. J. Permann, D. Andrs & J. M. Miller, Continuous integration for concurrent MOOSE framework and application development on GitHub. Journal of Open Research Software, 3(INL/JOU-15-34179). 2015.

[23] Ecured. Metodologías Agiles. https://www.ecured.cu/Metodolog%C3%ADas_Agiles

[24] D. Calvo, Metodologías Tradicionales vs Metodologías Ágiles. https://www.diegocalvo.es/metodologias-tradicionales-y-metodologias-agiles/metodologias-tradicionales-vs-metodologias-agiles/

[25] J. D. Gonzalez Alzate, Implementación de CI/CD en Kubernetes usando Kaniko y Tekton. 2021.

[26] J. Briceño, Propuesta De Despliegue Continuo Y Virtualización Basada En Contenedores Para Ambientes De Pruebas De Software. 2020. https://repository.udistrital.edu.co/bitstream/handle/11349/28005/PROYECTO%20FINAL%20-%20Barrera%20-%20Brice%C3%B1o.pdf?sequence=1

[27] A. L. Oviedo Briones, Estudio de las ventajas del manejo de cloud computing (computación en la nube) y propuesta de un modelo de uso para nuestro medio (Bachelor's thesis, QUITO/PUCE/2011). 2011.

[28] S. Tidwell, Argo CD vs Tekton vs Jenkins X: Finding the Right GitOps Tooling. 2021. https://platform9.com/blog/argo-cd-vs-tekton-vs-jenkins-x-finding-the-right-gitops-tooling/

[29] A. Lefort, Diferencias entre cloud y on-premise. 2019. https://cloud28plus.com/es/article/Diferencias-entre-Cloud-vs-On-Premise

[30] M. A. Mascheroni & E. Irrazábal, Problemas que afectan a la Calidad de Software en Entrega Continua y Pruebas Continuas. In XXIV Congreso Argentino de Ciencias de la Computación (La Plata). 2018.

[31] E. Uribe Ángeles, UNNE aporta soluciones para pruebas en el desarrollo de "software de entrega continua". 2021. Disponible en http://observatorio denoticias.redue-alcue.org/unne-aporta-soluciones-para-pruebas-en-el-desarrollo-de-software-de-entrega-continua/

[32] Servicenow, ¿Qué es la entrega continua? Disponible en: https://www.servicenow.es/products/it-operations-management/what-is-continous-delivery.html#:~:text=La%20entrega%20continua%20crea%20un%20entorno%20de%20trabajo,su%20trabajo.%20Entrega%20continua%20frente%20a%20despliegue%20continuo

[33] Ó. Pérez Galán, Automatización y control de despliegue de infraestructuras. 2018.

[34] J. Peralta Mori, DevOps en la entrega continua de la Oficina General de Estadística y Tecnología de la Información y Comunicaciones del Ministerio de Cultura, Lima. 2019.

[35] P. A. Castañeda García, Prácticas DevOps de entrega continua de software para la transformación digital de los negocios (Doctoral dissertation, Universidad EAFIT). 2019.

[36] 360 Logica, To Do, In Progress, Done Disponible en: https://www.360logica.com/blog/kanban-board-an-easy-approach-to-customize-your-work/capture_/

[37] C. Sánchez Gómez, Estudio de herramientas de despliegue continuo de aplicaciones, y sus ventajas competitivas en un mundo marcado por la agilidad (Bachelor's thesis). 2020.

[38] D. Fergon, De la nada al «Todo Continuo» y del Scrum más canónico al Kanban con #NoEstimates. 2015. Disponible en http://davidfergon.github.io/2015-06-16-de-la-nada-al-todo-continuo-de-scrum-a-kanban-y-noestimates

[39] D. Maggi, Diseño de una arquitectura basada en contenedores para la integración y el despliegue continuo (CI/CD). 2020.

[40] K. Nakamura, Azure DevOps YAML build pipeline: Where is checkout?.2020. https://dev.to/kenakamu/azure-DevOps-yaml-build-pipeline-where-is-checkout-3nci

[41] S. Muntaner Ruiz, Plataforma de integración y despliegue continuo.

[42] [CI/CD] Continuous Integration + Continuous Delivery + Continuous Deployment. (2019). Disponible en: https://luisiblogdeinformatica.com/ci-cd-continuous-integration-continuous-delivery-continuous-deployment/

[43] E. Suescún-Monsalve, C. J. Pardo-Calvache, S. A. Rojas-Muñoz & A. Velásquez-Uribe, DevOps in Industry 4.0: A Systematic Mapping. Revista Facultad de Ingeniería, 30(57). 2021.

[44] J. Freixas, La transformación digital de la mano del cloud computing y DevOps. 2022.

[45] DevOps – una nueva generación de developers [Internet]. Disponible en https://juancarlosabaunza.com/DevOps-una-nueva-generacion-de-developers/

[46] A. Élez Villamarín, Introducción a DevOps para la mejora de los procesos de desarrollo con herramientas Open Source.

[47] DevOps: Qué es y el caso de Netflix Disponible en https://www.quantion.com/es/2020/12/21/DevOps-que-es-y-el-caso-de-netflix/

[48] Barclays adoptó la cultura ágil de DevOps para mantener la competitividad Disponible en https://www.redhat.com/es/success-stories/barclays

[49] Double v partners Disponible https://doublevpartners.com/

[50] Movet disponible en https://movet.co

[51] Stoyco Disponible en https://stoyco.io/es/

[52] Caso de éxito del banco scotiabank colpatria disponible en https://www.scotiabankcolpatria.com/publicaciones-banco/news/caso-exito-google

[53] Davivienda transforma la cultura de la empresa y se vuelve 100% digital en el trabajo colaborativo Disponible en https://cloud.google.com/customers/banco-davivienda?hl=es-419