

DevOps como estrategia de aporte a la calidad de productos de software en MIPYMES
desarrolladoras en el contexto colombiano

Pablo Aníbal Bejarano De la Hoz ✉ pabhoz@gmail.com
Andrés Mauricio Heredia Guerrero ✉ andres.heredia@gmail.com

Tesis de Maestría presentada para optar al título de Magíster en Ingeniería de Software

Asesor: Hugo Armando Ordoñez Erazo, Doctor (PhD)
Ingeniería telemática



Universidad de San Buenaventura Colombia
Facultad de Ingeniería
Maestría en Ingeniería de Software
Santiago de Cali, Colombia
2019



Citar/How to cite [1]

Referencia/Reference [1] P Bejarano, A Heredia, H Ordoñez. (2019). “DevOps como estrategia de aporte a la calidad de productos de software en MIPYMES desarrolladoras en el contexto colombiano” (Tesis Maestría en Ingeniería de Software). Universidad de San Buenaventura Colombia, Facultad de Ingeniería, Cali.

Estilo/Style:
IEEE. (2014)



Bibliotecas Universidad de San Buenaventura



Biblioteca Digital (Repositorio)
<http://bibliotecadigital.usb.edu.co>

- Biblioteca Fray Alberto Montealegre OFM - Bogotá.
- Biblioteca Fray Arturo Calle Restrepo OFM - Medellín, Bello, Armenia, Ibagué.
- Departamento de Biblioteca - Cali.
- Biblioteca Central Fray Antonio de Marchena – Cartagena.

Universidad de San Buenaventura Colombia

Universidad de San Buenaventura Colombia - <http://www.usb.edu.co/>

Bogotá - <http://www.usbbog.edu.co>

Medellín - <http://www.usbmed.edu.co>

Cali - <http://www.usbcali.edu.co>

Cartagena - <http://www.usbctg.edu.co>

Editorial Bonaventuriana - <http://www.editorialbonaventuriana.usb.edu.co/>

Revistas - <http://revistas.usb.edu.co/>

AGRADECIMIENTOS

A nuestros asesores (y principalmente a nuestro director) que con gran paciencia nos instruyeron en el camino, a cada una de las empresas que participó de buena voluntad y abrió sus puertas apostando por una estrategia en desarrollo, al programa por su formación de calidad y apoyo constante, a cada una de esas personas que nos ofreció ánimo durante este largo camino.

Dedicamos este trabajo a nuestras familias que siempre han creído en nuestro trabajo e ideal de apostar por fomentar la inversión de esfuerzos en el producto nacional, y a todas las empresas MIPYMES que se encuentran en constante búsqueda de estrategias para ofrecer productos de calidad y un lugar de armonía y profesionalismo en el que trabajar.

RESUMEN

Los reportes de la industria colombiana de desarrollo de software indican un problema en la calidad de sus productos que resulta en bajos niveles de exportación, asociado además a otras malas prácticas de la industria. En este trabajo se presenta el diseño de una estrategia de implementación de DevOps que permite identificar los pilares que agrupan los habilitadores para la misma con el fin de validar el aporte a la calidad de los productos de desarrollo de software para las MIPYMES colombianas que se consigue.

Posterior a la implementación de la estrategia en 5 empresas se logra llegar hasta la última de las etapas con una de ellas y los resultados de la validación muestran que no sólo el objetivo de mejora de calidad es logrado, sino que otros aspectos reportados en los estudios también fueron positivamente influenciados.

Palabras clave: DevOps, Organizational culture, quality, continuous integration, continuous deployment, continuous delivery, continuous testing, MIPYMES, MSMEs.

ABSTRACT

Reports from the Colombian software development industry indicate a problem in the quality of its products that results in low export levels, associated with other industry malpractices. This paper presents the design of an implementation strategy of DevOps that allows to identify the pillars that group the enablers for it to validate the contribution to the quality of software development products for Colombian MSMEs that is achieved.

After the implementation of the strategy in 5 companies it is possible to reach the last of the stages with one of them and the results of the validation show that not only the objective of quality improvement is achieved, but other aspects reported in the studies were also positively inferred.

Keywords: DevOps, Organizational culture, quality, continuous integration, continuous deployment, continuous delivery, continuous testing, MIPYMES, MSMEs.

TABLA DE CONTENIDO

	PÁG.
I. INTRODUCCIÓN.....	10
A. PLANTEAMIENTO DEL PROBLEMA.....	12
B. JUSTIFICACIÓN.....	14
C. OBJETIVOS.....	15
1) Objetivo General	15
2) Objetivos Específicos	15
D. RESULTADOS OBTENIDOS	15
II. CONTEXTO TEÓRICO	16
A. AUTOMATIZACIÓN DE PROCESOS	16
1) Integración Continua	16
2) Entrega Continua.....	16
3) Despliegue Continuo	17
B. PRUEBAS DE SOFTWARE.....	17
1) Pruebas de Unidad.....	18
2) Pruebas de Integración	18
3) Pruebas Funcionales.....	18
4) Pruebas de Aceptación	18
5) Pruebas de Regresión	19
C. INFRAESTRUCTURA ELÁSTICA	19
1) Operación Continua.....	20
2) Contenedores	20
D. DEVOPS	21
1) Introducción	21
2) Características	22
3) Cultura Organizacional	24

DEVOPS COMO ESTRATEGIA DE APOORTE A LA CALIDAD DE PRODUCTOS DE SOFTWARE EN MIPYMES DESARROLLADORAS EN EL CONTEXTO COLOMBIANO

4) Tecnología e Infraestructura.....	26
5) Medición: Motivadores de DevOps	27
a) El éxito del negocio	28
b) La experiencia del cliente	29
c) El desempeño de la aplicación.....	30
d) La velocidad	30
e) La calidad	31
III. ESTRATEGIA PROPUESTA.....	32
A. Diagnostico Para La Transición Progresiva A DevOps	32
B. Definición De Metas, Objetivos Compartidos Por Los Integrantes De La Empresa Y La Razón Para Implementar DevOps	33
C. Diagnóstico De La Cultura Organizacional, Conocimientos Tecnológicos Y De Infraestructura	34
D. Diagnóstico De Compatibilidad Con La Tecnología, Prácticas Tecnológicas E Infraestructura Actual.....	36
E. Validación De La Implementación De La Cultura Organizacional, Prácticas Tecnológicas E Infraestructura Requerida	37
IV. EVALUACIÓN Y RESULTADOS.....	39
A. ETAPA 0. DEFINICIÓN DE METAS, OBJETIVOS COMPARTIDOS POR LOS INTEGRANTES DE LA EMPRESA Y LA RAZÓN DE IMPLEMENTAR DEVOPS	39
B. ETAPA 1. DIAGNÓSTICO DE LA CULTURA ORGANIZACIONAL Y CONOCIMIENTOS TECNOLÓGICOS Y DE INFRAESTRUCTURA.....	42
1) Distribución de departamentos propios y tercerizados	43
2) Metodologías usadas	45
3) Innovación e Investigación.....	46
4) Comunicación y Confianza	48
5) Conocimientos técnicos asociados a DevOps	51
6) Drivers de DevOps	53

DEVOPS COMO ESTRATEGIA DE APORTE A LA CALIDAD DE PRODUCTOS DE SOFTWARE EN
MIPYMES DESARROLLADORAS EN EL CONTEXTO COLOMBIANO

a.	Tiempo de respuesta ante situaciones críticas.....	53
b.	Tiempo de respuesta ante situaciones críticas.....	54
c.	Causas de fallas en productos de software.....	55
7)	Cierre de la etapa 1.....	56
C.	ETAPA 2. DIAGNÓSTICO DE COMPATIBILIDAD CON LA TECNOLOGÍA, PRÁCTICAS TECNOLÓGICAS E INFRAESTRUCTURA ACTUAL.....	57
D.	ETAPA 3. VALIDACIÓN DE LA IMPLEMENTACIÓN DE LA CULTURA ORGANIZACIONAL, PRÁCTICAS TECNOLÓGICAS E INFRAESTRUCTURA REQUERIDA.....	60
V.	CONCLUSIONES Y TRABAJOS FUTUROS.....	63
E.	CONCLUSIONES.....	63
F.	TRABAJOS FUTUROS.....	63

LISTA DE FIGURAS

	PÁG.
FIG. 1. MOTIVADORES Y MÉTRICAS DE DEVOPS	28
FIG. 2. CUADRANTE DE ESTADO PARA LA TRANSICIÓN A DEVOPS.....	35
FIG. 3. RESULTADO ENCUESTA JUNTA DIRECTIVA	40
FIG. 4. RESULTADOS ENCUESTAS MIEMBROS DE LAS EMPRESAS	41
FIG. 5. METODOLOGÍAS USADAS.....	45
FIG. 6. RECURSOS DESTINADOS A I+D+I.....	46
FIG. 7. PERCEPCIÓN DE INCENTIVOS PARA I+D+I	47
FIG. 8. NIVELES DE COMUNICACIÓN ENTRE LOS MIEMBROS.....	48
FIG. 9. NIVELES DE CONFIANZA ENTRE LOS MIEMBROS	49
FIG. 10. NIVEL DE CONOCIMIENTO TÉCNICO	51
FIG. 11. TIEMPO DE RESPUESTA ANTE EVENTOS	53
FIG. 12. CUMPLIMIENTO DE LAS ACTIVIDADES EN LA JORNADA LABORAL.....	54
FIG. 13. CAUSAS DE FALLA EN LOS PRODUCTOS DE SOFTWARE	55
FIG. 14. PRÁCTICAS EN UNA LAPS	57
FIG. 15. TECNOLOGÍAS Y SU COMPATIBILIDAD CON DEVOPS POR EMPRESA.....	58

LISTA DE TABLAS

	PÁG.
TABLA 1. PILARES PARA LA IMPLEMENTACIÓN DE DEVOPS	32
TABLA 2. VALORES CUANTITATIVOS Y CUALITATIVOS PARA LOS PILARES	35
TABLA 3. DEPARTAMENTOS EN LAS EMPRESAS	44
TABLA 4. DEPARTAMENTOS TERCERIZADOS EN LAS EMPRESAS	44
TABLA 5. PRACTICAS TECNOLÓGICAS POR EMPRESA.....	52
TABLA 6. RESULTADOS DE LA IMPLEMENTACIÓN DEVOPS EN LA EMPRESA B	60

I. INTRODUCCIÓN

El desarrollo tradicional de software dentro de las empresas se presenta como una distribución de esfuerzos descentralizados entre las distintas partes que trabajan en ella, siguiendo cada una objetivos específicos que finalmente deberían aportar a un objetivo general común; puntualmente en empresas dentro del sector del desarrollo de software los equipos suelen agruparse en dos grandes áreas (desarrollo y operaciones), trabajando de manera independiente y algunas veces hasta convirtiéndose en rivales internos, y aun así la industria del software se ha mantenido a flote con dicha estrategia, al menos hasta ahora, pues la capacidad de producción comienza a flaquear al momento de enfrentar las nuevas demandas de un mercado exigente que requiere cambios constantes de manera ágil, conservado altos estándares de la calidad del producto [1].

En consecuencia han surgido aproximaciones para dar una solución a esta problemática, tales como, el agilismo (metodologías ágiles) en el desarrollo de software, la integración continua, las herramientas de automatización, entre otras prácticas para el mejoramiento de los procesos de producción de software [1-4], la adopción de estas estrategias en las organizaciones suele presentarse de manera aislada y no logra el objetivo para el cual se realizó su implementación dentro de la empresa [5]; esto debido a la descentralización de las prioridades principales de cada uno de los equipos involucrados. Un ejemplo de ello, se puede evidenciar cuando la empresa determina implementar agilismo únicamente con los miembros del equipo de desarrollo, obteniendo como resultado la generación de releases de manera periódica, pero limitando el despliegue de los mismos a la capacidad de trabajo del equipo de operaciones, quienes enfocados en asegurar la estabilidad del entorno de producción, se ven obligados a frenar el despliegue; convirtiendo el trabajo entre desarrollo y operaciones en una relación conflictiva negativa que se aleja de las metas de la organización y no en una colaboración positiva que permita alcanzar los objetivos planteados. Aunque existe gran abundancia de métodos y herramientas, el éxito en el uso de las mismas presenta gran dificultad, y para procurar solucionar todas estas falencias surge el concepto de DevOps, que en estudios anteriores reporta mejoras en rendimiento y

calidad de software tales como: 200x en frecuencia de despliegues, 24x menos tiempo en recuperación de fallas, 3x en reducción de fallas originadas por cambios, reducción de costos, mejor respuesta a las necesidades de los clientes y hasta una mejor percepción de la empresa a nivel profesional incrementando su visibilidad en el mercado, entre otros [6-8].

A. PLANTEAMIENTO DEL PROBLEMA

Tras la revisión del estado de la industria del software en Colombia se puede evidenciar, que (1) al menos el 80% del sector TI es considerado una MIPYME (Micro, pequeñas y medianas empresas), de las cuales al menos el 50% ofrece productos y/o servicios de desarrollo de software, QA, IaaS y Cloud Computing [9], (2) que las exportaciones del país han disminuido mientras que la importación de software dobla el resultado de producto exportado según [10-11], los resultados anteriormente mencionados además estarían asociados a la falta de inversión en I+D+i, los niveles de productividad y la calidad de los productos de software, lo cual ubica el producto de software colombiano como no competitivo en el mercado según lo plantea [9, 12], que además afirman que la inversión en I+D+i impactaría las ventas, producción y calidad de los productos en un incremento de al menos 20 y 30%.

Entre la lista de las debilidades de la industria del software propuesta por [13] en su estudio del crecimiento de la industria del software en Colombia, se destacan (1) los productos de baja calidad, (2) los bajos niveles de innovación, y (3) la falta de organización empresarial; de esto sobresale que según [6] el 64% de incidentes negativos en la percepción del producto son atribuidos a la calidad del mismo. Como una posible solución a todos estos aspectos se plantea generar una estrategia para implementar DevOps en la empresa desarrolladora de software colombiana.

Para poder llevar a cabo la implementación adecuada de DevOps en cualquier empresa, primero es necesario tener clara la definición del concepto, y es en esto en donde se presenta una gran problemática, pues muchos autores presentan distintas definiciones del mismo como también lo afirman [14-15], por ejemplo [3] manifiesta la dificultad de la definición de DevOps como un concepto, y lo plantea como un rol comparado con los ingenieros de release, asociando tareas específicas dentro del proceso de producción de software al “ingeniero DevOps”, mientras [4, 16, 17] plantean DevOps como un conjunto de prácticas enfocadas en la agilización, optimización y calidad de entrega del software,

incluso [4] plantea las fases clave del desarrollo de software con DevOps: (1) planeación continua, (2) integración continua, (3) despliegue continuo, (4) pruebas continuas y (5) monitoreo continuo; coincidiendo con [18] que además va un poco más a fondo en las prácticas planteando una tabla de las mismas separada por actores; mientras autores como [2, 15, 19, 20, 21] van a más profundidad introduciendo el concepto de la cultura organizacional, planteando entonces DevOps como un conjunto de prácticas y conceptos que no solo se basan en las prácticas de desarrollo de software sino también en prácticas culturales, clasificadas como facilitadoras y bloqueadoras; siendo entonces DevOps un Framework conceptual, lo que también deja claro que DevOps no es una fórmula de oro que puede ser aplicada a cualquier contexto de la misma manera esperando los mismos resultados.

Con base en lo anterior en el presente proyecto de investigación se plantea la siguiente pregunta de investigación:

¿Cómo definir una estrategia para la implementación de DevOps como factor de reducción de errores y tiempos de despliegues en MIPYMES desarrolladoras en el contexto colombiano para aportar a la calidad de productos de software?

Para darle respuesta a la anterior pregunta de investigación, se optó por orientar esta investigación al diseño de una estrategia DevOps como el aporte a la calidad de productos de software en MIPYMES colombianas; el cual será medido a través del uso de métricas basadas en calidad de pruebas de software tales como: Deployment Success Rate y Application Error Rate.

B. JUSTIFICACIÓN

Para combatir la falencia en calidad de productos de software (que como se menciona anteriormente, es una de las causas que más impactan la capacidad de exportación a la industria colombiana de desarrollo de software) muchas empresas han visto como posibilidad la implementación del framework DevOps (que además entre sus características intrínsecas combate problemas como la innovación, tiempos de respuesta y releases, entre otros), pues su promesa de venta es un conjunto de resultados ideales que en algunos casos podrían ser vistos hasta como una utopía, pero con una dificultad de implementación implícita que requiere un verdadero conocimiento del mismo para lograr el éxito, dicha implementación requiere de una clara comprensión del mismo e inversiones variables de recursos de las empresas (tiempo, dinero, RRHH, entre otros) basados en el estado actual de la misma con respecto a los pilares del framework. Se requiere de una preparación para siquiera contemplar la implementación, pues un conjunto de prácticas tecnológicas y su respectiva suite, no son más que una parte del todo y lastimosamente podrían ser el enfoque tomado por desconocimiento del framework. En este sentido, este trabajo propone una estrategia de implementación del framework que inicialmente cubre los aspectos de diagnósticos en las áreas clave requeridas para la misma, abordando el pilar de prácticas culturales que no pueden cubrir las suites ofertadas en el mercado, además dicha estrategia separa a manera de fases la preparación de la empresa, permitiendo evaluar la posibilidad de una implementación parcial o total del framework (acorde a la capacidad actual de la empresa misma y sus miembros), esto generaría en un proceso ágil con metas definidas que plantea a las directivas la posibilidad de detenerse sin pérdidas en una inversión sobre un alcance no contemplado, o más bien hasta un punto en el que los resultados de los diagnósticos y la disposición de las empresas deseen llegar. De esta manera cada empresa pueda adoptar las prácticas que generen valor en sus procesos de desarrollo acorde a sus necesidades.

C. OBJETIVOS

1) Objetivo General

Diseñar una estrategia de Implementación de DevOps como aporte a la calidad de productos de software en MIPYMES desarrolladoras en el contexto colombiano.

2) Objetivos Específicos

- Definir los factores que permiten la implementación de DevOps en un grupo de desarrollo pertenecientes a una MIPYMES.
- Implementar el diseño de la estrategia de DevOps basada en los factores identificados previamente en grupos de desarrollo pertenecientes a MIPYMES.
- Evaluar el impacto de la implementación con base a métricas basadas en pruebas de calidad de software, tales como: *Deployment Success Rate* y *Application Error Rate*.

D. RESULTADOS OBTENIDOS

- Capítulo de libro: Lineamientos para un acercamiento a la implementación de DevOps para MIPYMES en Colombia. Obras colectivas en ciencias de la computación, Págs. 505 - 511. ISBN 978-958-5415-19-5 [48].
- Estrategia para la implementación de DevOps en las MIPYMES colombianas. La estrategia planteada demuestra aportes a la calidad del producto de software dentro del contexto de la MIPYMES desarrolladora de software en Colombia, entre otros aportes.
- Documento de investigación y resultados de esta.
- Un módulo de DevOps impartido dos veces durante cada versión del diplomado “Entornos de aplicaciones web” en el marco de FormaTIC.

II. CONTEXTO TEÓRICO

A. AUTOMATIZACIÓN DE PROCESOS

La automatización de procesos es una práctica del desarrollo de software que busca la reducción de errores operativos basándose en la capacidad de automatizar dichos procesos, como por ejemplo lo podrían ser los procesos de revisión de componentes de software, la integración entre ellos y hasta la misma la puesta en producción de los mismos. Entre los procesos automatizados requeridos para el pilar tecnológica de DevOps se encuentran los siguientes:

1) Integración Continua

Una de las prácticas de desarrollo de software contempladas dentro del pilar tecnológico de DevOps, que consiste en la automatización del proceso común de los desarrolladores en el que se compilan los componentes que han sido enviados al SCM central (Source Code Manager, en español sistemas de control de código fuente) y posteriormente se ejecutan las pruebas unitarias que los mismos requieran, en este caso de manera automatizada, esto con el fin de minimizar los posibles errores de procedimiento por la repetición humana de los procesos, encontrar también defectos en componentes de software de manera temprana para tomar medidas correctivas y optimizar los tiempos de validación de los mismos para garantizar mejores entregas.

2) Entrega Continua

Es una de las prácticas de desarrollo de software contempladas dentro del pilar tecnológico de DevOps, la entrega continua consiste en el proceso de la preparación de los componentes para convertirse en artefactos listos para producción, lo cual implica básicamente que dentro de su proceso se ha contemplado la integración continua. Durante la entrega continua, se llevan a cabo procesos automatizados de pruebas de las

compilaciones que van más allá de las pruebas de unidad, Incluyen pruebas como lo son las de integración, carga, regresión, aceptación, entre otras, con el fin de realizar una validación integral del *release*, que será ubicado usualmente en un gestor de artefactos de software para la posterior aprobación manual y entrega a producción por el encargado del proceso. La finalidad de la entrega continua entonces extiende la misma de la integración continúa adicionando las ventajas de automatizar proceso de publicación de software únicamente delegando la aprobación y entrega del *release* al encargado del proceso de forma manual.

3) Despliegue Continuo

Una de las prácticas de desarrollo de software contempladas dentro del pilar tecnológico de DevOps, el despliegue continua es muy similar a la entrega continua pero con una ligera variación, en este caso el despliegue en producción del *release* se realiza también de manera automatizada, lo que concluiría en la automatización total de la línea de producción de software, lo cual sería ideal para un proceso muy maduro que requiere de la mayor celeridad para la liberación del producto o de constantes actualizaciones.

B. PRUEBAS DE SOFTWARE

Las pruebas de software son una buena práctica que aporta a la calidad del desarrollo de componentes de software mediante la verificación del buen funcionamiento de los mismos, siendo estas ejecutadas de manera manual o automatizada tanto por los mismos desarrolladores como por los miembros de equipos de aseguramiento de la calidad, y aunque actualmente se conocen las bondades del proceso de automatización de las mismas, los números en las encuestas continúan siendo muy bajos [22]. A continuación, se expondrán algunas de las pruebas de software más populares en la industria [23] desde lo planteado por [24]:

1) Pruebas de Unidad

Nuevamente ganando popularidad entre los tipos de pruebas comparando los resultados desde 2012 [25] hasta 2018 [23], las pruebas de unidad consisten en validar el correcto funcionamiento de unidades de software, siendo esta lo más pequeña posible, si la unidad posee dependencias, estas deberán ser simuladas con un comportamiento esperado para el buen funcionamiento de la unidad a probar, a estas simulaciones se les conoce como *stubs*. Se considera a las pruebas de unidad el tipo de prueba más básica en el proceso.

2) Pruebas de Integración

Consisten en la verificación de la funcionalidad de varias unidades del sistema trabajando en conjunto, estas pruebas se realizan con la finalidad de comprobar si las unidades individuales también logran su cometido de manera conjunta.

3) Pruebas Funcionales

Consisten en la verificación de alguna funcionalidad del sistema en concreto, similar a las pruebas de aceptación, pero con un enfoque desde la operación del sistema en lugar de las interacciones del usuario.

4) Pruebas de Aceptación

Las pruebas de aceptación describen los escenarios desde el punto de vista del usuario simulando la interacción de este con el sistema, esto con el fin de realizar un recorrido desde la perspectiva del usuario por el sistema para validar que los criterios de aceptación para el escenario se cumplen a cabalidad. Algunas suites de pruebas incluso ejecutan el navegador y permiten visualizar el proceso de la interacción.

5) Pruebas de Regresión

Las pruebas de regresión consisten en volver a ejecutar las (pruebas) previamente ejecutadas en el momento en que una actualización a alguna unidad o la adición de una nueva se lleva a cabo, esto con el fin de verificar que dicha unidad no afecta el comportamiento deseado del sistema, en el caso de ser así, se aplicará una regresión, lo que restablecerá el sistema al último estado óptimo existente. Las pruebas de regresión no siempre requieren la ejecución de todas las pruebas previas del sistema, en ocasiones puede requerir solamente la de las unidades asociadas al caso de uso del elemento en cuestión.

C. *INFRAESTRUCTURA ELÁSTICA*

Junto a las necesidades de un crecimiento rápido bajo la demanda de los usuarios consumidores de servicios, crecen también los retos del planteamiento de una infraestructura capaz de soportar cargas intensas durante periodos críticos, situación en la que hasta empresas del tamaño de Yahoo manifestaron la necesidad de una solución eficiente frente al escalamiento basado en máquinas virtuales [26], a este nuevo reto en el que empresas como Amazon se destacaron como proveedoras de servicios de tipo IaaS (Infraestructura como servicio) se le conoce como arquitecturas elásticas, de hecho, uno de los productos estrella del conjunto de servicios de AWS (Amazon Web Services) es la famosa nube elástica de la mano de las EC2 por sus siglas que en inglés corresponden a Elastic Cloud Computing, dichas máquinas permiten realizar lo que se conoce como infraestructuras elásticas, las cuales se fundamentan en la capacidad de crecimiento de infraestructura bajo demanda con el fin de garantizar la prestación del servicio con la potencia requerida para finalmente volver a su estado básico cuando ya no sea requerida la potencia alcanzada, esto entonces se convierte en otro factor clave en cuanto a los costos de mantenimiento de infraestructura y adquisición de capacidades computacionales [27].

1) Operación Continua

La operación continua consiste en la capacidad de siempre estar funcionando independientemente de los procesos de mantenimiento y actualización, esto fundamentado en las capacidades de las arquitecturas elásticas y las tarifas cómodas ofrecidas por servicios IaaS comparadas con la adquisición de componentes propios. Un ejemplo de la práctica de la operación continua es la actualización de un nodo mientras otra instancia del mismo se encuentra atendiendo las peticiones de los usuarios, para posteriormente regresar a producción con el nodo actualizado, a este tipo de prácticas se les agrega la capacidad de pruebas ágiles del producto como lo hace Netflix con las nuevas funcionalidades, midiendo su aceptación para determinar si realizan una regresión o en su lugar aplican la misma a los demás nodos; Netflix también realiza pruebas para probar su capacidad de soporte a fallas en la operación mediante lo que ellos mismos denominaron Chaos Monkey [28], que al ser ejecutado se encarga de generar picos, apagar nodos y demás actividades que puedan generar un downtime al servicio, el cual está fundamentado en la ingeniería del caos, que consiste de manera concreta en generar situaciones de turbulencia en operación para probar la confiabilidad del servicio, la cual tiene ya una creciente comunidad dedicada a la formulación de las mismas [29].

2) Contenedores

Con la necesidad de mejorar el uso de los recursos de infraestructura y la cantidad de estos que la virtualización completa de un sistema operativo supone, una medida acertada en el uso de recursos y aprovechamiento de los mismos es la que plantea la virtualización de servicios mediante contenedores. Un contenedor es una unidad estandarizada de software usada para empaquetar el código y las dependencias de aplicaciones para portar dichas entre diferentes entornos. Los contenedores aíslan su contenido del entorno que los aloja, pero comparten el kernel del sistema operativo del

host evitando que cada aplicación desplegada por medio de contenedores requiere de un sistema operativo individual dentro de los recursos que usan. Actualmente el estándar de la industria definido para los contenedores fue creado por Docker [30].

D. DEVOPS

1) Introducción

Mientras las empresas solicitan habilidades abrumadoras para la contratación de personal capacitado para llevar a cabo el rol de un ingeniero de DevOps [3] en medio de la incertidumbre de lo que es; deberían primero centrar esfuerzos en el esclarecimiento de DevOps y el objetivo de implementarlo.

Definir DevOps es mucho más complejo que identificar sus beneficios, los cuales son muy claros en casos y reportes como [7, 31, 39], pero llegar a un acuerdo en su definición ha tomado muchos rumbos. Autores como [3] plantean primero el siguiente cuestionamiento: “¿Es DevOps un nuevo rol o un nuevo equipo de trabajo dentro de la organización?” y se preguntan cómo la academia está formando a los ingenieros que asumirán este rol, pregunta que se ha trabajado en [32]; mientras tanto, otros autores plantean DevOps como un conjunto de prácticas que buscan asegurar la calidad del software mientras se reduce el tiempo de entrega entre cada uno de los cambios que el mismo requiera [4, 33]. Algunos autores justifican que la complejidad en la definición de DevOps recae en que su implementación en cada organización no es genérica, es decir, DevOps no es una receta de éxito que se aplica por igual, sino que debe ser adoptada según cada caso [14], y algunos otros lo clasifican como un framework [2; pero entre un mar de definiciones la que se destaca en éste estudio es la que trasciende todas las anteriores, definiendo DevOps no sólo como un conjunto de prácticas y tecnologías, sino como el conjunto de una filosofía de trabajo empresarial que plantea un vuelco a la cultura empresarial de desarrollo tradicional introduciendo el concepto de cultura colaborativa [34]. DevOps es un Framework conceptual de trabajo que

implica una reestructuración general de la cultura de las partes implicadas en los procesos de la empresa, lo que no sólo involucra a los equipos de desarrollo y operaciones, sino a la empresa como un único individuo trabajando en pro de los beneficios que el framework conlleva. DevOps propone la eliminación de la barrera del desarrollo tradicional que separa organizacionalmente las tareas y responsabilidades, generalmente en los equipos de desarrollo y operaciones, proponiendo la reintegración de los mismos en pro de combatir el problema del cuello de botella en la entrega del software y la velocidad de la implementación de los cambios. Algunas de las causantes que generan la brecha entre estos dos departamentos son (1) que el equipo de desarrollo se inclina por el trabajo ágil en el desarrollo de características y corrección de errores, mientras el equipo de operaciones se focaliza en la estabilidad del sistema; intereses que no van alineados a un objetivo en común, (2) las métricas de desempeño y rendimiento diametralmente opuestas aplicadas al área de desarrollo vs las aplicadas a operaciones y (3) la cultura organizacional plantea que cada departamento es responsable de sus propios intereses y debe limitarse a los mismos, en lugar de trabajar adoptando una cultura de colaboración.

El objetivo principal entonces de DevOps es el de reducir los tiempos que existen entre los cambios del software y la puesta en producción de este asegurando aun así la calidad planteada [35], unificando en una meta en común los objetivos específicos de cada equipo, trabajando de manera colaborativa.

2) Características

DevOps se plantea para mitigar problemas comunes de la industria de software moderna como lo son: (1) el constante miedo al cambio, tanto a nivel tecnológico como en el aspecto de la evolución del mismo software debido al riesgo que implican los mismos, (2) el engorroso proceso de despliegue que lleva consigo la constante incertidumbre que sólo cesa cuando todo se lleva a cabo y nada (al menos explícito)

salta a la vista como un error y por último y más común (3) el problema de la delegación de culpas, en donde el no cumplimiento de las metas de cada equipo es justificado por el bloqueo que otro equipo le representa [36].

DevOps es caracterizado por su agilidad e innovación, partiendo del hecho de que el desarrollo tradicional plantea a los equipos de la empresa como silos separados, impidiendo el escalamiento de la misma, impone filosofías y motivaciones opuestas a los equipos (un ejemplo de ello es la búsqueda de cambios ágiles de desarrollo que va en contraposición con la conservación de la estabilidad de operaciones) y le teme a la adopción de nuevas tecnologías en su afán de conservar un proceso estable, perdiendo la posibilidad de evolución [17, 31].

DevOps está compuesto por aspectos claves como: (1) la cultura, (2) la automatización, (3) la medición, (4) la colaboración [14]; incluso algunos autores adicionan a la lista el (5) monitoreo [2]. La adopción de estos aspectos claves requiere la toma de acciones y la identificación de los factores que alientan o no las prácticas para la adopción. [14] plantea que la adopción implica (1) remover la frontera entre equipos de desarrollo y operaciones asignando responsabilidades distribuidas a los ingenieros que pertenecen a los mismos, (2) reforzar la comunicación entre los equipos y unificar los protocolos para que la integración sea más sinérgica, claro está, sin necesidad de desaparecer la diferenciación de los roles de cada miembro, y (3) propone la creación de un equipo que se encargue de conectar los dos equipos como una especie de equipo de control de DevOps.

- La cultura: DevOps requiere de una cultura colaborativa centrada en la comunicación inspirada en el agilismo, que unifique las filosofías de cada equipo y las enfoque en el trabajo en búsqueda de un fin común.
- La automatización: DevOps requiere de un enfoque a la automatización de todos los procesos posibles para permitirse la agilidad que plantea, delegando menos tiempo de los equipos en procesos que no lo requieran.

- La medición: este proceso se realiza mediante la medición del esfuerzo implicado en el proceso de desarrollo de software basado en datos tomados en tiempo real, generando estimaciones de esfuerzo que reflejan una aproximación real, entregando mediciones más efectivas.
- La colaboración: se requiere un replanteamiento de la manera en que los equipos trabajan y más importante aún, se comunican entre ellos; la comunicación efectiva y el conocimiento de los procesos mutuos incentiva la agilidad y los buenos resultados en la obtención de las metas establecidas.
- El monitoreo: implica la constante revisión de los procesos de desarrollo y el sistema como tal (tarea que en el desarrollo tradicional se delega a las operaciones), permitiendo obtención automática de datos de rendimiento y desempeño de los sistemas desarrollados.

DevOps requiere de planeación continua, integración continua, despliegue continuo, pruebas continuas y un monitoreo continuo llevados a cabo como una colaboración entre desarrollo y operaciones, lo cual permitirá la identificación de mayor información relevante de seguimiento [4-5].

Una vez Identificado y aplicado DevOps correctamente se esperaría entonces que el proceso de desarrollo de software tuviera las capacidades de: (1) planeación continua, (2) despliegue continuo y colaborativo, (3) pruebas e integración continua, (4) monitoreo y optimización continuo, (5) recuperación rápida de fallas y (6) retroalimentación continua del comportamiento de los usuarios [34].

3) Cultura Organizacional

El éxito en la adopción de DevOps en una organización depende en gran medida de la cultura organizacional, como lo enuncia [34, 37]. Existen rasgos característicos que permiten evidenciar la buena adopción de DevOps en la organización, como (1) metas y objetivos compartidos, (2) la necesidad de colaboración, (3) la innovación como

bandera, (4) la experimentación como fuente de aprendizaje (a partir de sus éxitos y aceptación de las fallas), (5) un entorno en el que los miembros se sientan parte de un gran equipo que les brinde soporte y apoyo ante las diferentes situaciones y (6) una comunicación transparente y basada en la empatía. Otros son valores de la cultura que son tan genéricos a cualquier organización pero que deben existir para eliminar toda fricción entre los miembros del equipo tales como confianza, respeto y responsabilidad compartida [34]. Uno de los requisitos a cumplir es que la organización tenga claridad en por qué adoptar DevOps, dado que esta claridad permite ser punto de apoyo para establecer estrategias que permitan superar los múltiples impedimentos que pueden surgir en el proceso de transición, (1) una estructura organizacional poco o nada orientada al agilismo y a la colaboración, (2) resistencia al cambio, (3) recarga de trabajo de operaciones en los desarrolladores y operaciones, en vez de una mejora en la productividad y la eficacia, (4) apatía en la adopción de competencias blandas (colaboración, comunicación, adaptabilidad) [36] [38].

El cambio cultural que DevOps facilita la identificación de metas en común entre equipos y por ende estimula la colaboración para actividades de esfuerzo en conjunto, beneficiando a la organización [14] en lugar del clásico modelo en que cada miembro de su equipo se encargaba de sus propias responsabilidades, asemejando sus labores como dice [17] a la entrega de sus resultados arrojándolos sobre el muro que les divide y desentendiéndose de los mismos.

Para garantizar que la adopción de DevOps en su empresa no se vea afectada se debería entonces tener gran cuidado, la claridad en la definición de las razones y metas por las cuales se lleva a cabo este proceso. La estructura organizacional debe ser replanteada de manera colaborativa y por supuesto, también es necesario comprender que DevOps no siempre será la solución adecuada para todos los requerimientos de sus clientes, como podría ser el caso en que los procesos ágiles de fallo y error no son aceptables, esto y algunos aspectos profundos de los puntos críticos para procurar la salud del aspecto cultural de la organización son tratados en [34].

4) Tecnología e Infraestructura

El modelo de DevOps plantea aprovechar las bondades de los servicios en la nube, tales como: (1) la infraestructura como servicio (IaaS), (2) la plataforma como servicio (PaaS) y (3) el software como servicios (SaaS) para permitirse la elasticidad que la agilidad intrínseca del mismo requiere [38], lo que implica a su vez dejar a un lado el temor a las nuevas tecnologías y entender que es necesaria una transición como plantea [16]. Una vez implementado lo planteado anteriormente se introduce el concepto de microservicios, que permitirán una verdadera definición arquitectural que soporte tecnológicamente DevOps [33, 37].

Los beneficios de la adopción de DevOps incluyen: (1) la reducción de tiempos en procesos, (2) la reutilización tanto de infraestructura como de software, (3) la escalabilidad, (4) la calidad y agilidad de entregas, entre otros beneficios [4].

Para permitirse adoptar DevOps, la estrategia tecnológica de la empresa debe alinearse con (1) la automatización de procesos, como fue discutido previamente en éste documento, (2) infraestructura elástica altamente escalable, para lo cual autores como [1, 19, 32, 37, 39, 40, 41, 42, 43] proponen el uso de tecnologías en la nube, servidores virtuales y contenedores como Docker, reduciendo tiempo en procesos partiendo de la premisa de que lo que ya está hecho no hace falta volver a ser realiza; y (3) una bien definida línea base de la configuración tanto de código como infraestructura [34]. De lo anterior que las empresas podrían ajustar las necesidades tecnológicas y de infraestructura a los valores financieros que puedan permitirse (pues se dispone de muchas herramientas de bajo presupuesto como se evidencia en [37, 42]) y adoptar exitosamente (desde el área tecnológica y de infraestructura) siempre que su estrategia esté bien definida y se vele por garantizar los siguientes aspectos:

La gestión de la configuración

La gestión de la calidad en tiempo real y con mediciones aterrizadas por la retroalimentación como se habla en [4-5].

La automatización de la infraestructura.

Algunos estudios muestran entre sus resultados cuáles deberían ser las habilidades ideales para los miembros que afrontarán el reto de aplicar DevOps como [3, 14, 39], mientras otros se enfocan desde el planteamiento de modelos para implementar DevOps según las necesidades de las organizaciones [5] hasta definiciones del framework [36, 42, 44], estandarizaciones [39] y el acercamiento de la adopción de estándares vigentes relacionados al cloud computing como lo es TOSCA (Topology and Orchestration Specification for Cloud Applications) para la estandarización de artefactos [1, 42].

5) Medición: Motivadores de DevOps

A medida que DevOps es comprendido y adoptado por la organización, se debe tener especial cuidado en alinear sus metas y objetivos con las del negocio, de manera tal que se pueda percibir la verdadera naturaleza de su implementación y razón de ser en la organización. DevOps más que una entrega de software más rápido, con menos errores, con una cultura de colaboración y menos conflictos entre los desarrolladores y operaciones, es en sí, un canalizador para el entendimiento del porqué hacemos las cosas, hacia dónde va el negocio y cómo esas actividades aportan a lograr esa meta.

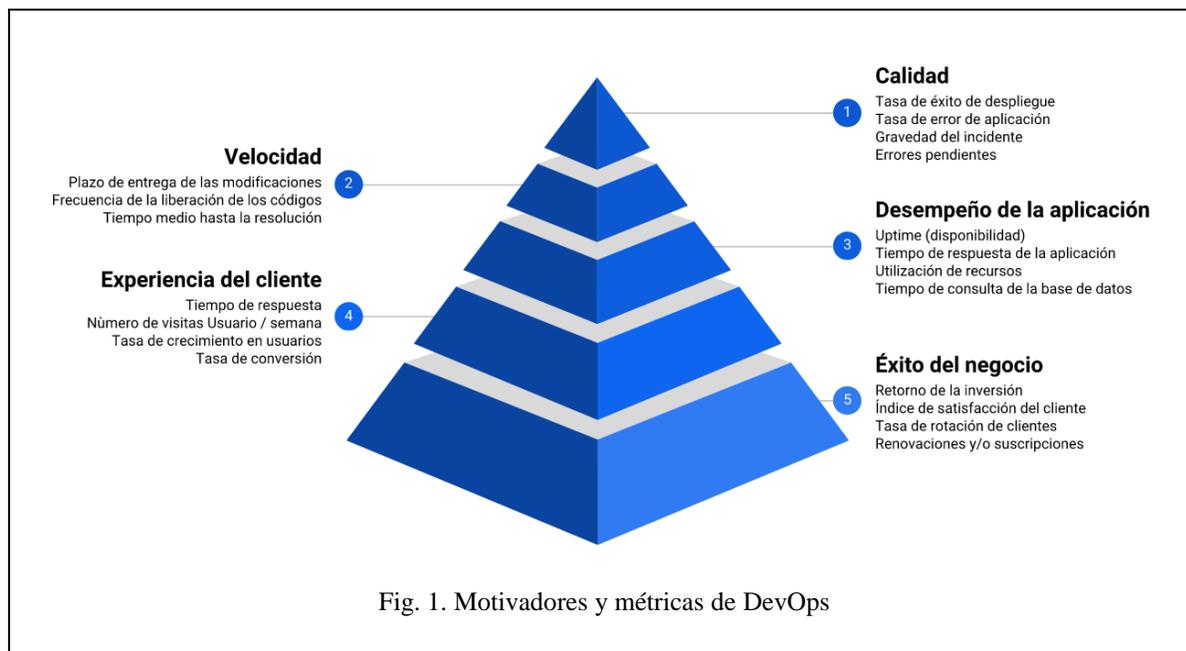


Fig. 1. Motivadores y métricas de DevOps

Según [12] DevOps cuenta con cinco motivadores críticos para la medición de su efectividad en la implementación, ver fig. 1, se describen a continuación:

a) *El éxito del negocio*

El software genera valor cuando se puede relacionar a un conjunto de metas del negocio, que se pueden medir, tales como, el índice de satisfacción del cliente, la tasa de conversión de potenciales clientes en clientes, ingreso promedio por cliente, costo de adquisición de clientes, entre otras.

Para desarrollar o mantener software que genere valor para la organización, es crítico que esté alineado con los objetivos de negocio, de manera tal, que esta alineación se convierta en un factor de medición, que permita ayudar tanto al equipo de desarrollo, como al negocio, a entender la relación de lo que hacen mediante un conjunto de indicadores, los cuales dependen del tipo de industria en el que la empresa se encuentre, estos motivadores pueden cambiar, por ejemplo en el caso del entretenimiento y SaaS podría considerarse una métrica

las tasas de conversión, las renovaciones o las nuevas suscripciones, así como también, en algunos casos, la satisfacción del cliente a través de entregas tempranas y continuas de software de valor

La mayoría de estas métricas son del ámbito financiero y de negocio y algunas de ellas son:

- Retorno de la inversión
- Índice de satisfacción del cliente
- Tasas de conversión
- Promedio de ingresos por usuario
- Costes de adquisición de clientes
- Tasa de rotación de clientes
- Ingresos recurrentes o incrementales
- Renovaciones y/o suscripciones

b) La experiencia del cliente

No basta con tener un producto innovador, alineado con el negocio y que genere valor, si funcionalmente falla, los clientes rápidamente abandonarán el producto, si es complicado, confuso, tiene muchos pasos en las transacciones importantes, puede generar frustración al usuario. Se deben llevar métricas de indicadores claves de rendimiento (KPI) para entender la experiencia del cliente y prevenir la fuga, para obtener datos para mejorar la dicha experiencia y para conocer si se está haciendo bien.

- Algunos indicadores serían los siguientes:
- Tiempos de respuesta percibidos de las transacciones clave
- Frecuencia de las transacciones clave
- Número de visitas por usuario/ por semana

- Tasas de crecimiento de los usuarios
- Tasas de conversión
- Cantidad de tiempo que se pasa en la aplicación
- Resultados de las pruebas A/B

c) El desempeño de la aplicación

Teniendo un software alineado con el negocio y cuya experiencia de usuario es positiva, debo entonces medir y establecer KPI's de rendimiento, disponibilidad, escalabilidad, tiempos de respuesta, entre otros, de manera que la experiencia del usuario no se vea afectada por temas técnicos. Y no menos importante, poder responder de manera rápida ante los eventos que ocurran en producción.

- Uptime (disponibilidad)
- Tiempo de respuesta de la aplicación
- Tiempo de respuesta de la base de datos
- Porcentaje de tiempo de transacción en la base de datos
- Utilización de los recursos
- Tiempos de consulta de la base de datos

d) La velocidad

La implementación con miras a ganar velocidad en el desarrollo, la entrega y el tiempo que toma corregir los errores que se presenten en producción es uno de los enfoques de DevOps, estas son algunas métricas que pueden ayudar a registrar este progreso, sin perder de vista la calidad del producto entregado y en los cuales por lo general se apoya en la Infraestructura como servicio y la nube pública, siempre teniendo en cuenta los motivadores anteriores:

- Plazo de entrega de las modificaciones
- Frecuencia de la liberación de códigos
- Tiempo medio hasta la resolución

e) La calidad

Entregar de manera rápida y periódica es importante, pero más importante aún es entregar con calidad, en este punto la Integración Continua (CI), Despliegue continuo, Entrega continua y las métricas relacionadas a continuación cobran una gran importancia, sobre todo cuando se pueden medir pre y post despliegue:

- Tasa de éxito de despliegue
- Tasa de error de aplicación
- Gravedad del incidente
- errores pendientes

Los anteriores motivadores se deben adaptar al tipo de negocio e industria que tenga la organización, y se deben monitorear y registrar (generar y mantener los indicadores o KPI's) y su proceso de adopción puede realizarse de manera incremental, siendo factible concentrarse en estabilizar el rendimiento de la plataforma, para luego incrementar la innovación y la alineación con los objetivos de negocio.

III. ESTRATEGIA PROPUESTA

Las actividades identificadas según los resultados encontrados en la investigación indican que se debe (1) identificar prácticas culturales, e (2) identificar prácticas tecnológicas de la empresa para hacer un diagnóstico del estado actual de la misma en ambas, y así posteriormente (3) definir una estrategia de implementación de DevOps con base a dichas prácticas (Ver Anexo 4).

De toda la recopilación presentada se han identificado dos pilares primordiales a tener en cuenta para una implementación de DevOps, los cuales están descritos a continuación:

TABLA 1. PILARES PARA LA IMPLEMENTACIÓN DE DEVOPS

Cultura Organizacional	Tecnología e Infraestructura
<ul style="list-style-type: none"> a) Metas y objetivos compartidos b) Colaboración c) Innovación d) Mejora continua e) Confianza f) Comunicación 	<ul style="list-style-type: none"> a) Automatización de procesos <ul style="list-style-type: none"> I) Integración continua II) Entrega continua III) Despliegue continuo IV) Pruebas continuas b) Infraestructura elástica <ul style="list-style-type: none"> I) Operación continua II) Contenedores

A. Diagnostico Para La Transición Progresiva A DevOps

La adopción de DevOps requiere de una inversión de dinero y esfuerzo variable, directamente relacionada con el estado de las prácticas culturales, tecnológicas y de infraestructura de cada empresa, y para determinar dicho estado es necesario haber comprendido el framework y realizar un diagnóstico propio con respecto a los pilares previamente identificados, lo cual usualmente no se lleva a cabo por múltiples razones, entre las más populares: (1) se desconocen los aspectos de la autoevaluación a aplicar, (2) no existe tiempo para realizar una autoevaluación, (3) no existe dinero para la adquisición

de infraestructura y tecnología necesaria, y (4) los miembros de los equipos muestran rechazo a salir de su zona de confort, por ello se ha diseñado esta propuesta para el diagnóstico y diseño para la transición progresiva a DevOps, que permita a las empresas avanzar minimizando los riesgos mediante la definición de fases y lineamientos de una estrategia adecuada al negocio específico de las mismas, permitiéndoles un progreso ajustado a sus tiempos y presupuestos, atacando de manera aislada cada una de las partes claves.

Como primera parte de la estrategia, es necesario que la empresa realice un proceso de definición de metas y objetivos compartidos de la misma (identificada dentro de la propuesta planteada como: “0. Definición de metas, objetivos compartidos por los integrantes de la empresa y la razón para implementar DevOps”), con los cuales cada miembro de los equipos podrá alinear la misión y objetivos de su equipo con los compartidos por la empresa, de allí que cada uno de los miembros podrá también identificar su rol dentro de su equipo y con ello los aspectos a los que su trabajo aporta para la consecución de metas y objetivos tanto personales, como de equipo y globales. Una vez realizado el proceso, la empresa deberá socializar con cada uno de los miembros de los equipos el framework DevOps, y establecer cuáles son sus intenciones o motivaciones para la transición al mismo, pues esto permitirá la retroalimentación de las opiniones de los miembros y dará un punto objetivo al que llegar con la transición, con el cual se podrá realizar la verificación del impacto.

B. Definición De Metas, Objetivos Compartidos Por Los Integrantes De La Empresa Y La Razón Para Implementar DevOps

Tener claro el propósito de una organización es vital para conocer cómo orientarla al éxito, así mismo las empresas requieren definir la razón o razones para implementar DevOps, de tal manera que tanto el proceso como los resultados puedan ser contrastados con dicha definición para revisar si se han alcanzado los objetivos de la misma, pero para ello primero se deben tener claras las metas de la empresa, tanto por la junta directiva como por cada miembro de la misma, que al saberlo, podrá revisar los objetivos de su rol dentro

de la organización y orientar su labores a un trabajo colaborativo en búsqueda de las metas conjuntas.

Durante esta investigación se encontró que muchos de los empleados desconocen aún el propósito de su cargo actual, y las metas de la empresa, su misión y visión, dichos empleados entonces no podrían enfocar sus esfuerzos a un trabajo colaborativo si ni siquiera reconocen a totalidad sus deberes en la organización.

En esta etapa se plantea la socialización y validación del conocimiento de los miembros de la empresa de la misión y visión de esta, los roles existentes y sus deberes dentro de la organización y por último la razón por la cual se planea la implementación de DevOps, con el fin de que cada participante del proceso conozca cuál es su responsabilidad y aporte en la misma.

C. Diagnóstico De La Cultura Organizacional, Conocimientos Tecnológicos Y De Infraestructura

Para el diagnóstico del estado de las prácticas culturales y el conocimiento con respecto a las prácticas tecnológicas y de infraestructura, se desarrolló un test que debía ser diligenciado por todos los involucrados de la empresa, dicho test se apoyó en múltiples test de innovación y prácticas empresariales[13][46][47], y busca generar un diagnóstico del nivel en que la empresa se encuentra con respecto las competencias blandas (en sus prácticas culturales) y el conocimiento global de los miembros de cada equipo con respecto al pilar de tecnología e infraestructura, determinando la ubicación del estado de la empresa (obtenido por la evaluación de los datos del diagnóstico) en el cuadrante presentado en la figura 2 propuesto por los autores de este trabajo.

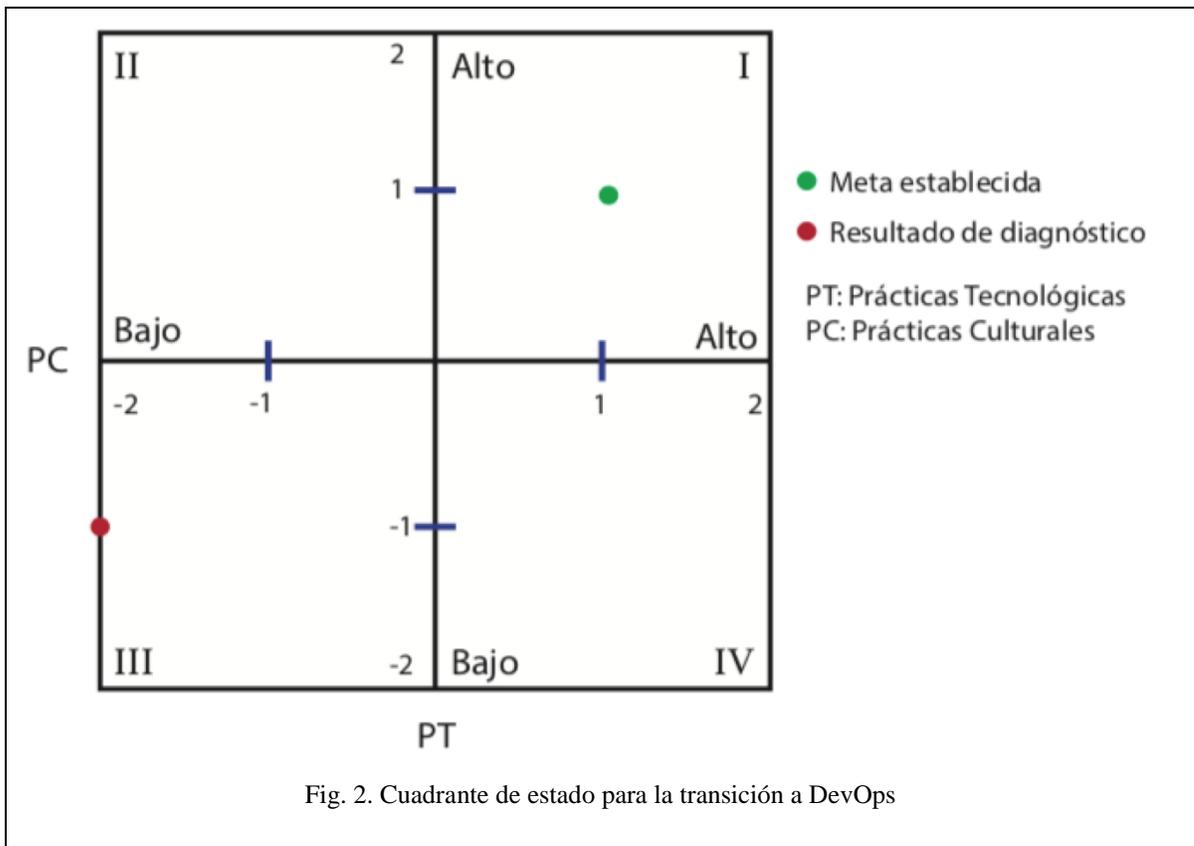


Fig. 2. Cuadrante de estado para la transición a DevOps

Según el resultado obtenido en el diagnóstico, la ubicación de la empresa podrá estar comprendida entre el cuadrante I y IV del plano y éste representará el resultado en el test diagnóstico de las prácticas tecnológicas y de infraestructura (PT) y las prácticas culturales (PC), siendo los posibles resultados: (1) muy habilitado, (2) habilitado, (3) requiere mejores condiciones, y (4) no habilitado, dichos valores corresponden al resultado cuantitativo obtenido en el análisis, a continuación se describirán las asociaciones de los valores cuantitativos y cualitativos para cada pilar:

TABLA 2. VALORES CUANTITATIVOS Y CUALITATIVOS PARA LOS PILARES

1.5 – 2.0	Muy habilitado
1.0 – 1.4	Habilitado
-1.0 – 0.9	Requiere mejores condiciones
< -1	No habilitado

Tomando como ejemplo el caso presentado en la Fig.2, dicha empresa obtuvo un resultado numérico de menos uno (-1) en PT y menos dos (-2) en PC, dicho resultado surge de la tabulación de cada una de las preguntas realizadas en el test diagnóstico, las cuales tienen asignadas un peso relativo al nivel de impacto dentro del estado para la transición compuesto entre dos (un impacto muy positivo) y menos dos (un impacto muy negativo), según lo anterior, la empresa estaría ubicada en el cuadrante III concluyendo que en cuanto a PT, es necesario mejorar las condiciones para continuar con la transición, mientras que en PC no se encuentran habilitados y se tendría que trabajar en la generación de prácticas culturales, es allí entonces en donde basados en los lineamientos definidos, se presenta un plan de mejora para atacar las oportunidades de mejoras evidenciadas en el informe resultante de la evaluación del diagnóstico, dicho informe es compartido con los directivos, los cuales deberán determinar si están dispuestos continuar con el proceso de transición e implementar el plan de mejora, o si por lo contrario desean detener la transición en este punto para evitar invertir más esfuerzos en ello. Si la respuesta obtenida es positiva (si desean continuar), la estrategia debe ser implementada y posterior a ello se aplicará de nuevo el test diagnóstico 1, buscando obtener los resultados establecidos como meta por la empresa, siempre que esta meta debe estar ubicada en el cuadrante I del plano del estado para la transición.

D. Diagnóstico De Compatibilidad Con La Tecnología, Prácticas Tecnológicas E Infraestructura Actual

Para el diagnóstico de compatibilidad con la tecnología se planteó una revisión del estado del pilar tecnológico de la empresa, esto con el fin de evaluar la compatibilidad de sus tecnologías y prácticas tecnológicas con los elementos requeridos identificados en el correspondiente pilar de DevOps.

Inicialmente se abordan las tecnologías usadas, con el fin de verificar la compatibilidad, pues en un caso negativo es requerido por la empresa sopesar la posibilidad de un cambio de la misma, para los casos positivos en la intención del cambio se debe plantear la nueva posibilidad a elegir basada en la compatibilidad con el pilar tecnológico, por otro lado para

los casos negativos, la empresa puede plantearse concluir con su proceso de implementar DevOps, o procurar una implementación parcial, la cual no garantizará los resultados propuestos, y que igualmente requerirá de una verificación del impacto de la misma en los resultados esperados. En los casos positivos a la compatibilidad será necesario entonces pasar a evaluar el estado de las prácticas, las cuales pueden ser “conocidas sin implementar”, “conocidas parcialmente implementadas” y “conocidas e implementadas”, no aplica la categoría “desconocidas” debido a que eso hace parte de uno de los objetivos del diagnóstico anterior; la finalidad de esta etapa es la de garantizar que la empresa pueda obtener una valoración en el nivel “conocidas e implementadas”.

Aunque el pilar tecnológico plantea cuatro prácticas en la categoría de automatización de procesos y dos en la de infraestructura elástica como base para DevOps, no quieren decir (1) que son las únicas que podrían ser implementadas y (2) que tengan todas que ser implementadas en todos los casos, pues depende del modelo de negocio algunas no son viables, como es el caso por ejemplo de productos de extrema responsabilidad como los médicos que deberían ser verificados finalmente por un experto de negocio antes de ser desplegados en producción, para dichos casos las prácticas de automatización de procesos hasta el nivel de una entrega continua serían ideales, y aun así, podrían existir casos en los que la empresa determine que su negocio debería implementar las prácticas de automatización hasta el nivel de integración continua, eso es una determinación que cada empresa debe sopesar durante del proceso.

E. Validación De La Implementación De La Cultura Organizacional, Prácticas Tecnológicas E Infraestructura Requerida

Como última etapa y parte primordial de la estrategia se plantea la validación de la implementación de DevOps dentro de la empresa para obtener información con respecto a los resultados obtenidos por la misma, esto como parte del proceso transversal de DevOps de mejora continua, el monitoreo permite la identificación de oportunidades de mejora que al ser atendidas adecuadamente permitirán la obtención de los objetivos deseados (los cuales fueron planteados desde la primera etapa en el proceso “0. Definición de metas,

objetivos compartidos por los integrantes de la empresa y la razón para implementar DevOps”).

Para este estudio puntualmente se realizó la medición de mejoras en términos de calidad de los productos de las empresas participantes para concluir si la implementación de DevOps mediante la estrategia planteada aportó a la calidad de estos.

IV. EVALUACIÓN Y RESULTADOS

Para el proceso de implementación se trabajó con la ayuda de cinco empresas desarrolladoras de software de la ciudad de Cali, cada una con situaciones distintas en cuanto a longevidad, enfoque, tecnologías y tamaño.

A. ETAPA 0. DEFINICIÓN DE METAS, OBJETIVOS COMPARTIDOS POR LOS INTEGRANTES DE LA EMPRESA Y LA RAZÓN DE IMPLEMENTAR DEVOPS

El primer acercamiento con las empresas consistió en una reunión de socialización de la propuesta y la solicitud de información por parte de la junta directiva de la misma con respecto a la misión y visión de la empresa, los roles que hacen parte de ella y sus descripciones, sus proyecciones y objetivos, todo esto de manera individual en el caso (en que la junta estuviese conformada por más de una persona), con el fin de posteriormente socializar lo compartido por cada miembro y de presentarse discrepancias, se pudiera unificar cada concepto antes de ser socializado con el resto de los miembros de la empresa.

Las preguntas realizadas fueron:

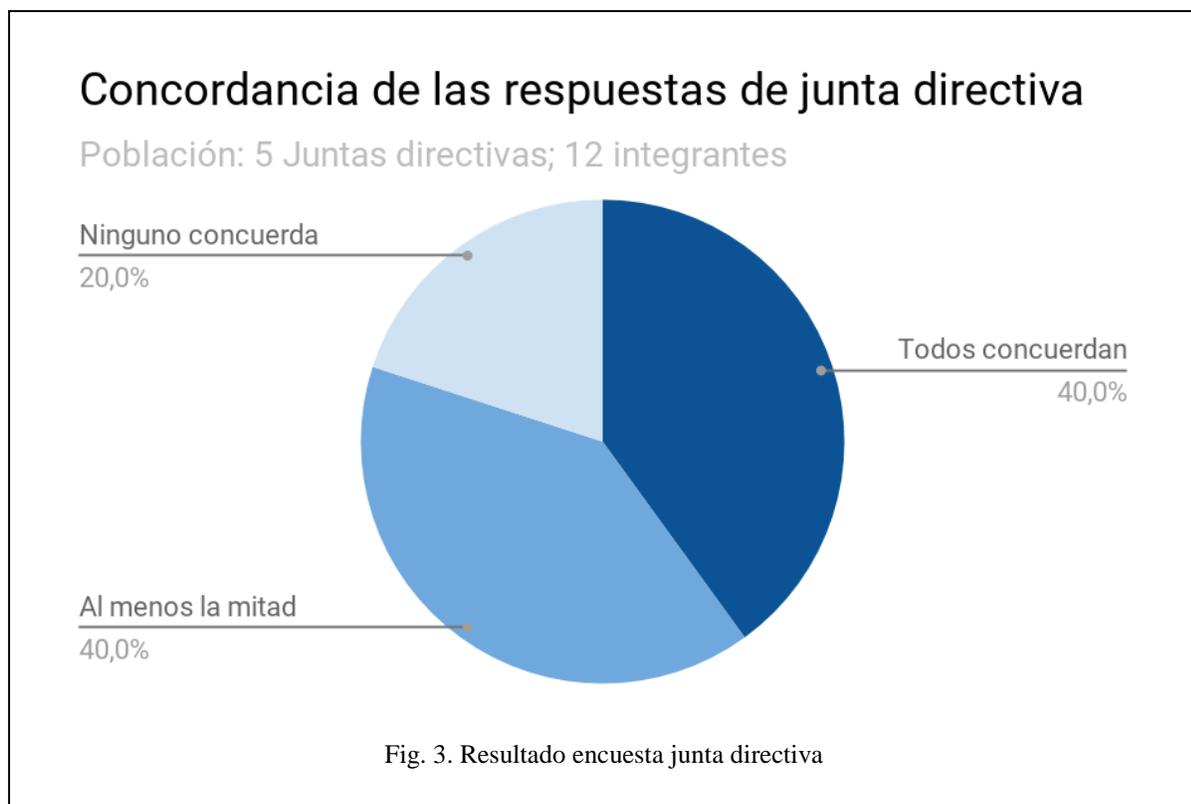
- ¿Cuál es la misión y visión de la empresa?
- ¿Cuáles son los objetivos de la empresa?
- ¿Cuáles son las metas de la empresa?
- ¿Qué roles existen dentro de su empresa? Descríbalos
- Teniendo en cuenta las respuestas anteriores ¿Por qué estamos trabajando todos?
- ¿Cuál es mi rol en lo anterior?
- ¿Qué es DevOps?
- ¿Por qué queremos implementar DevOps?

Estas mismas preguntas fueron extendidas a los miembros de la empresa para contrastar su punto de vista y verificar la claridad en cada uno de los aspectos solicitados.

Seguido a la recolección la información tanto de la junta directiva como de los empleados, se verificó qué tan consistentes eran las respuestas entre sí (de la misma junta directiva y posteriormente de los empleados con base a la definición unificada provista por la junta), todo esto como etapa 0 de la estrategia permitiendo con esas entradas generar las siguientes salidas:

- Definición de metas y objetivos compartidos para orientar la estrategia
- Definición inicial de roles y responsabilidades
- Definición de la motivación de la estrategia

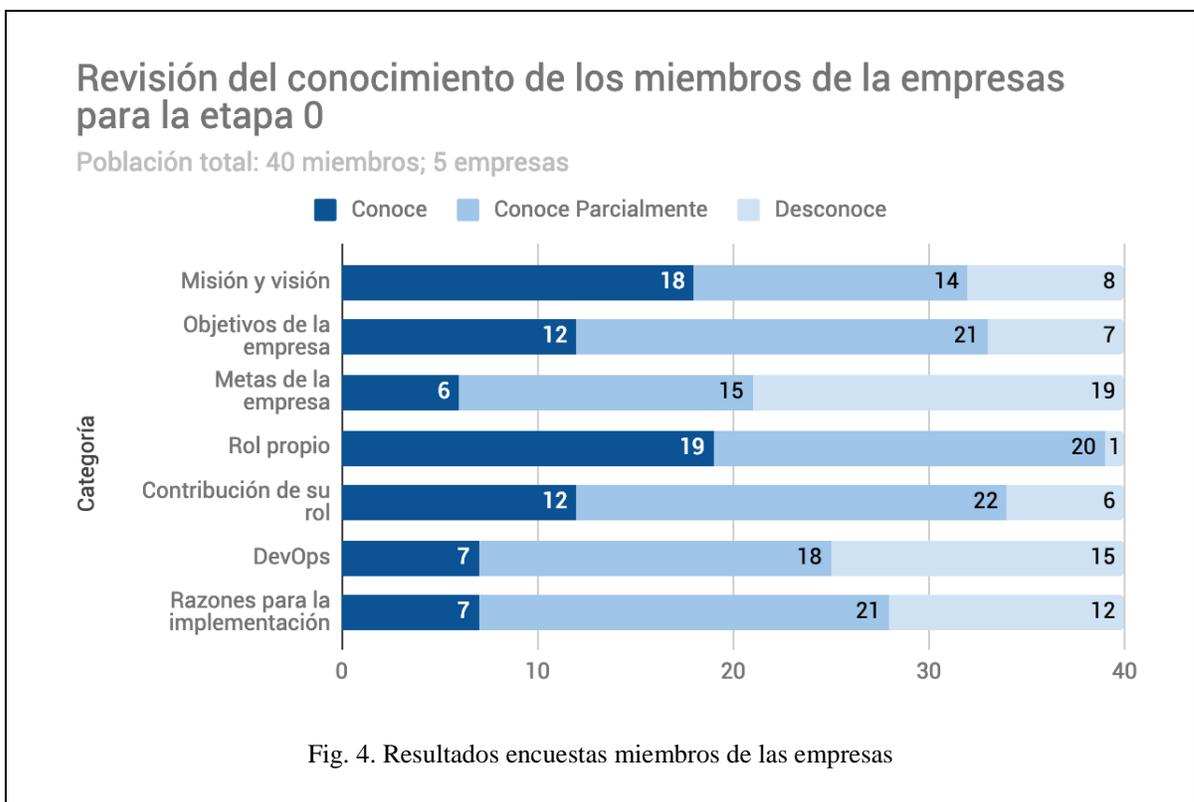
Los resultados obtenidos para esta etapa en las cinco empresas muestran que:



El 60% de las juntas directivas participantes no se encontraban totalmente alineadas en sus respuestas y por ende, primero se procedió a solucionar este inconveniente antes de continuar, para estos casos se socializo el resultado y se llevó a cabo una sesión de

unificación de conceptos, de tal manera que la población total de las juntas directivas lograron entrar en la categoría “Todos concuerdan”, además se revisó la claridad del concepto DevOps y las razones para la implementación del mismo dentro de las empresas, con esto se identificaron los objetivos a los que debería apuntar cada estrategia de implementación.

De las respuestas obtenidas por parte de los miembros de las empresas se obtuvo lo siguiente, fig. 4:



El conocimiento parcial y desconocimiento de los elementos básicos para iniciar el proceso de manera generalizada en las empresas expuso un punto crítico de capacitación de los miembros de cada una antes de iniciar con la primera etapa (en promedio ni siquiera EL 50% de la población conocía realmente las categorías consultadas), ver Fig. 4, para ello se programaron jornadas de socialización que permitieron entonces que la población total participante por cada una de las empresas se ubicara al estado “Conoce” en cada una de las

categorías, esto mediante la generación de material informativo facilitado por comunicación interna y reuniones. Como práctica empresarial se recomendó a cada empresa socializar con todos los miembros los aspectos anteriormente mencionados al menos de forma semestral y procurar campañas de comunicación que estén recordando los objetivos planteados y retroalimentando con los avances a los mismos. Durante las actividades los miembros de los equipos manifestaron sentirse muy a gusto con este tipo de socialización de información pues afirman conocer mejor su empresa, identificar su rol y el impacto de su trabajo en los objetivos de los demás roles y en el alcance del objetivo de su empresa.

B. ETAPA 1. DIAGNÓSTICO DE LA CULTURA ORGANIZACIONAL Y CONOCIMIENTOS TECNOLÓGICOS Y DE INFRAESTRUCTURA

Concluida la etapa anterior, es momento de ubicar a las empresas dentro del cuadrante correspondiente a su estado para la transición a DevOps apoyado en el instrumento de diagnóstico, primeramente, socializando a las empresas en qué consiste el test y qué se esperaba encontrar en los resultados obtenidos. La prueba fue efectuada en línea con el apoyo de los formularios de Google y la recolección de los datos se realizó de manera anónima, pero identificando el cargo de cada usuario. Cada prueba se realiza de manera independiente con cada empresa y tanto el instrumento como los resultados se manejaron de manera independiente.

El instrumento cuenta con cuatro (4) que son correspondientemente:

- Identificación del miembro.
- Diagnóstico empresarial (Estado y cultura empresarial).
- Prácticas tecnológicas.
- Percepciones de la empresa (Competencias blandas, innovación y empoderamiento).

Una vez recolectadas las respuestas se procedió al proceso de tabulación de las mismas y la ubicación dentro del cuadrante correspondiente (ver Anexos 1, 2, 3 y 5), y se responde a

la empresa con sus resultados y recomendaciones (las cuales también hacen parte de los anexos). Una vez realizada la ubicación de la empresa dentro del cuadrante y la identificación de sus oportunidades de mejora, se procedió a la reunión con las mismas para coordinar planes de mejora que permitieran mover la ubicación del cuadrante a un punto ideal definido previamente como la meta de la implementación, durante este proceso se trabajó de la mano con cada una en el desarrollo de estrategias e incluso se capacitó al personal de estas.

De los resultados recolectados se destacan:

1) Distribución de departamentos propios y tercerizados

Teniendo en cuenta los principios del framework la existencia del departamento y/o su tercerización pueden considerarse habilitadores o deshabilitadores, por ello casos como por ejemplo no contar con equipo propio de infraestructura aun cuando el departamento está tercerizado, puede afectar los tiempos de respuesta y la sinergia de los equipos como es el caso de la Empresa D, entre los departamentos cuya ausencia más desfavorecen la adopción del framework se encuentran: TI - Desarrollo de software, TI - Infraestructura y operaciones, TI - QA (Calidad); mientras por el contrario los departamentos que más aportan al proceso de transición serían: TI - Desarrollo de software, TI - Infraestructura y operaciones. Las tercerizaciones, aunque no son grandes penalizadores al proceso usualmente ofrecen una resistencia intrínseca debido a que cada empresa maneja sus propias políticas, ahora después de la tabulación de los resultados, ver tabla 3 y tabla 4, se obtiene que en este aspecto las empresas A, D y E obtienen un resultado positivo e igual (aunque mejorable), mientras que B y C son respectivamente menos favorecidas por sus condiciones.

TABLA 3. DEPARTAMENTOS EN LAS EMPRESAS

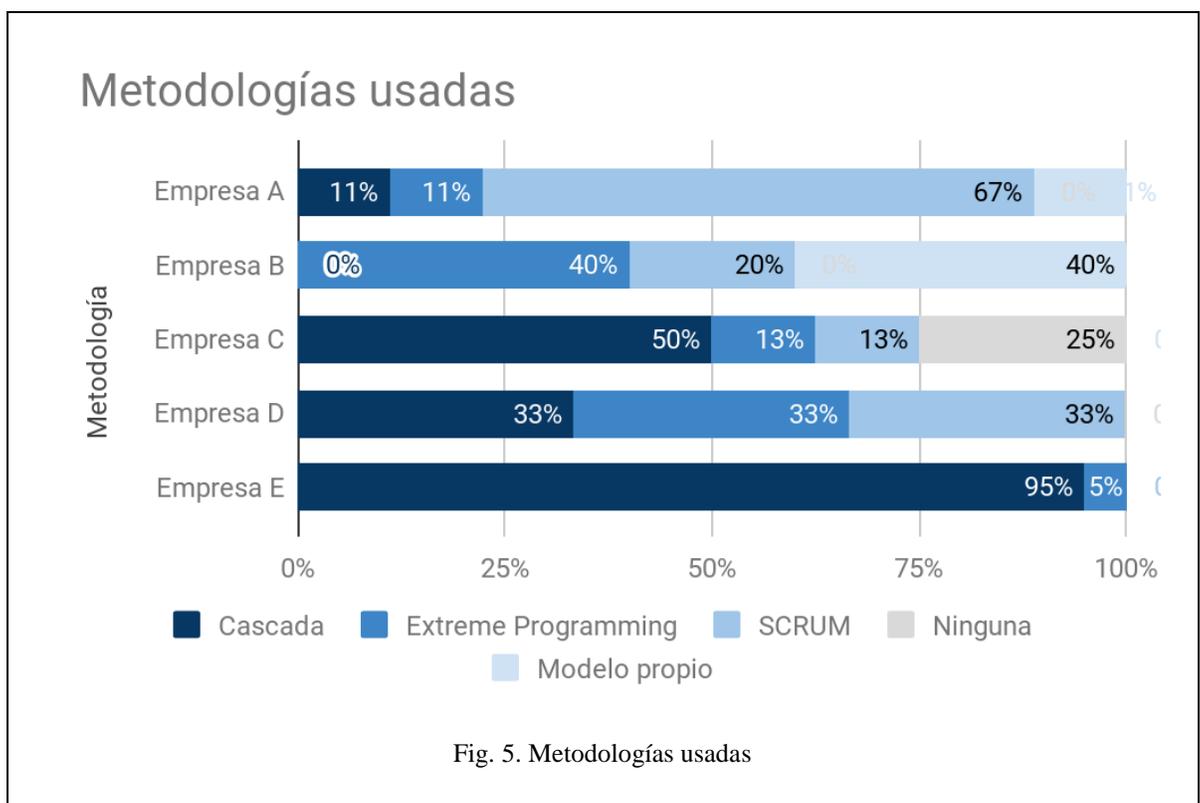
Departamentos con los que cuenta la empresa					
Departamento	Empresa A	Empresa B	Empresa C	Empresa D	Empresa E
TI - Desarrollo de software	X	X	X	X	X
TI - Infraestructura y operaciones	X	X			X
TI - QA (Calidad)				X	
TI - Seguridad	X				
TI - Gerencia	X	X	X	X	X
RRHH	X			X	X
Auditoría				X	
Ninguno de los anteriores					

TABLA 4. DEPARTAMENTOS TERCERIZADOS EN LAS EMPRESAS

Departamentos tercerizados					
Departamento	Empresa A	Empresa B	Empresa C	Empresa D	Empresa E
TI - Desarrollo de software					
TI - Infraestructura y operaciones	X	X		X	
TI - QA (Calidad)	X		X		
TI - Seguridad					
TI - Gerencia					
RRHH			X		
Auditoría	X	X	X		X
Ninguno de los anteriores					

2) Metodologías usadas

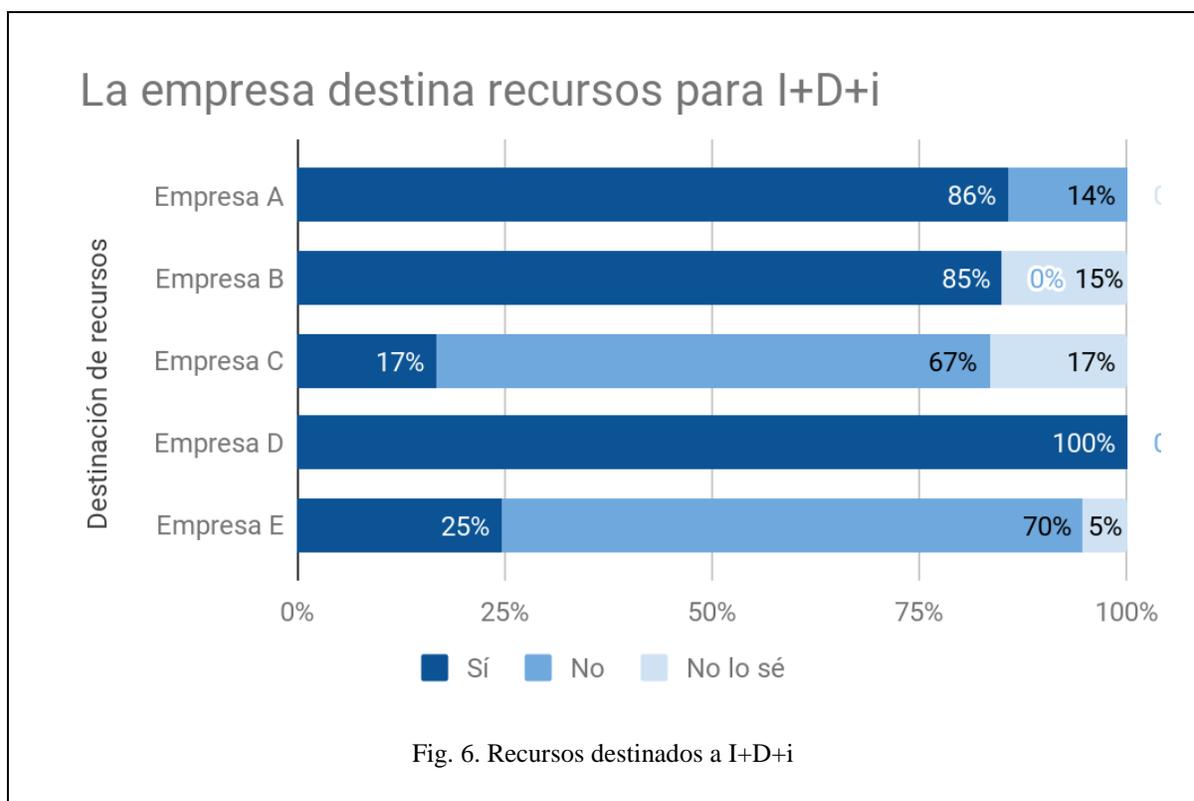
El agilismo como parte del ADN del framework plantea de manera implícita que la adopción de metodologías acordes favorecerá a la transición, aunque la adopción de ninguna metodología es aún más nociva que el uso de una no ágil. A continuación, se presenta el porcentaje de utilización de metodologías dentro de las empresas participantes:



Es evidente, ver Fig. 5, que la presencia de las metodologías tradicionales sigue siendo impactante en las empresas, esto dado a que no existe aún una visión de DevOps, con la cual progresivamente dichas metodologías se irán dejando a un lado hasta minimizarlas o desaparecer. Teniendo en cuenta la discusión previa el proceso de transición podría presentar más resistencia al cambio en las empresas C y E, mientras D y A presentan muy buenas condiciones y B se encuentra en un estado ideal con respecto a este apartado.

3) Innovación e Investigación

Siendo uno de los factores críticos en el desarrollo de valor por las empresas de desarrollo de software, es uno de los ejercicios que aún la visión tradicionalista no ha podido integrar dentro de su día a día y que está directamente relacionado con la disminución de la exportación de software colombiano como se evidencia en [36, 45].



Según los resultados obtenidos, Fig. 6, el 60% de las empresas en proceso logran la percepción de sus miembros en inversión para I+D+i, en donde nuevamente empresas como A, B y D se destacan alineados en algunas buenas prácticas habilitadoras para el proceso de adopción de DevOps, mientras C y E con una marcada tendencia tradicional nuevamente demuestran tener inconvenientes con ello. Ahora, si bien la destinación de recursos es vital, la motivación de los miembros para los procesos de

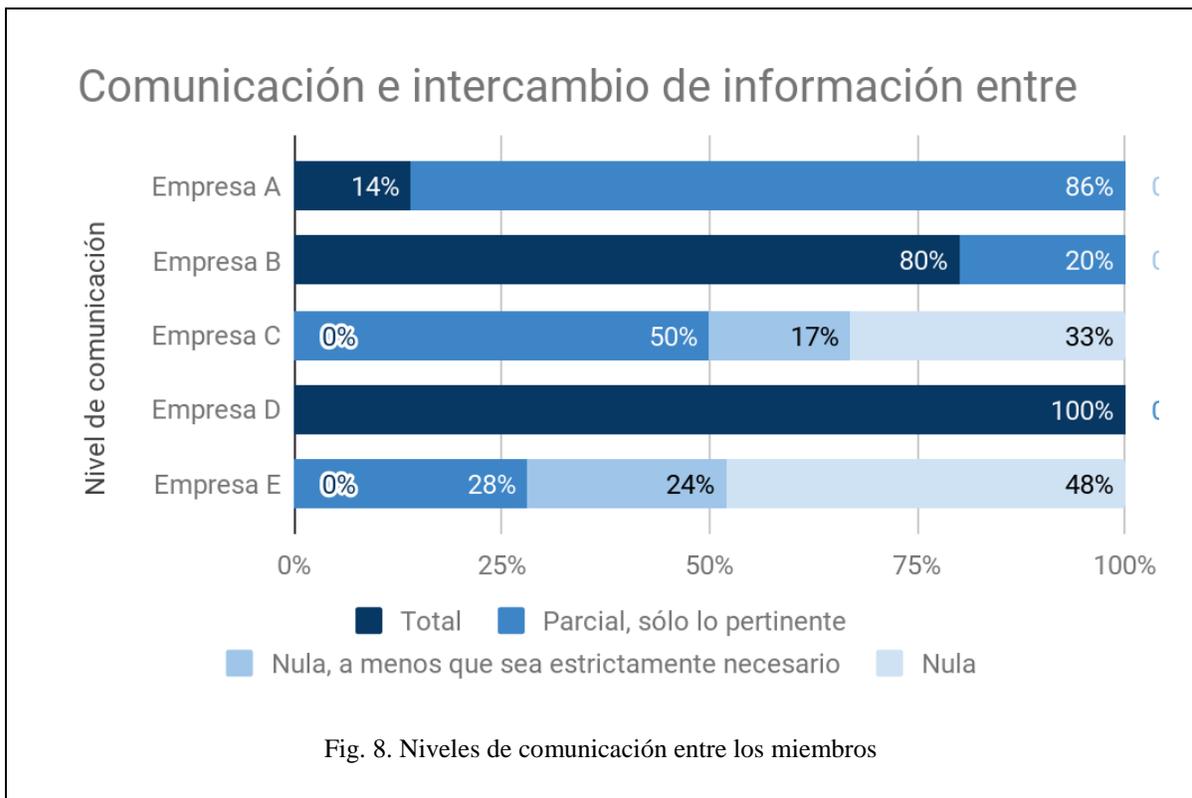
I+D+i no es irrelevante y a continuación se presenta la percepción que tienen ellos dentro de sus empresas al respecto:



Los resultados (Fig. 7) exponen entonces que casos como A en donde si bien los miembros manifiestan que existe una destinación de recursos para I+D+i, responden negativamente a la cantidad de estímulos que obtienen por ello y esto resulta en casos de apatía por asociar estos procesos con cargas adicionales que no generan ningún valor agregado para ellos, afectando la calidad de las actividades y los resultados; mientras tanto B y D se mantienen en una buena percepción tanto en destinación de recursos como en incentivos para lo mismo, mientras C y E quienes no tienen una buena respuesta a la pregunta de la inversión en I+D+i, además no demuestra una motivación del equipo para hacerlo.

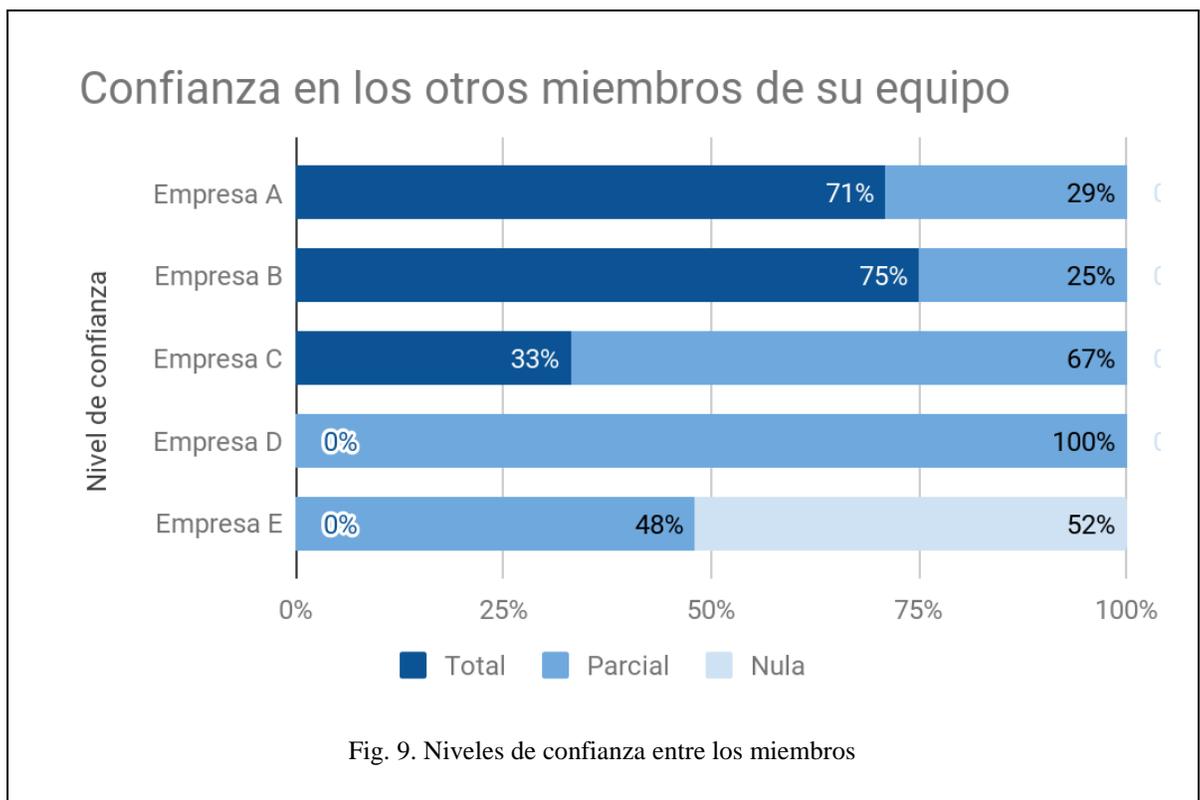
4) Comunicación y Confianza

Siendo competencias blandas muy relevantes en lo que a DevOps concierne, la comunicación y la confianza son conceptos complejos de dominar en entornos que no se perciben como propicios para ambas, adicionando a las empresas colombianas el agravante de una cultura del provecho y el utilitarismo, anclas arraigadas de las que es necesario desprenderse en pro del trabajo en equipo, la armonía y la sinergia. A continuación, se presentan los resultados obtenidos en cuanto a la percepción de la comunicación y la confianza de los miembros de cada empresa:



Para la interpretación de la figura 8 se establece que todo resultado por fuera del nivel de comunicación “Total” es una falencia en comunicación, los resultados de comunicación parcial son aceptables aun siendo una falencia en cuanto a que el trabajo sería enfocado en reforzar la comunicación, pero resultados de comunicación nula como lo son los niveles de comunicación representados en colores claros, son un gran

indicador de una oportunidad de mejora crítica que requiere la creación de canales y la incentivación de la misma. Los resultados entonces muestran una fortaleza en comunicación en las empresas B y D, por lo cual es un aspecto ya cubierto para continuar, mientras que para A se plantearon estrategias de refuerzo en la comunicación tanto entre equipos como los miembros mismos, y por último C y E quienes presentaron oportunidades de mejora críticas en este aspecto, requirieron de la revisión de canales actuales de comunicación y la creación de nuevos, además de la inclusión de la comunicación como pilar en sus procedimientos metodológicos, esto en búsqueda de la creación de una necesidad de comunicación efectiva.



Si siguiendo el planteamiento de la interpretación anterior, figura 9, la tonalidad oscura en los resultados identifica una percepción del nivel de confianza entre los miembros y el equipo de total confianza, mientras que las tonalidades más claras representan niveles parciales y nulos de confianza. Los resultados obtenidos muestran que la empresa B, quien anteriormente demostró niveles altos de comunicación, también tiene

una alta percepción de confianza total entre los miembros de la misma, permitiendo que cada uno trabaje sin guardar preocupaciones con respecto al desempeño de sus compañeros de manera tal que la sinergia se favorece enormemente, una buena comunicación y confianza permiten a cada individuo enfocarse en cumplir con sus responsabilidades, conocer las necesidades de los demás y aportar a ellas, todo esto sin realizar reprocesos por desconfianza en el trabajo de sus colegas; mientras tanto los resultados de D fueron una sorpresa, evidenciando que a pesar de contar con unas buenas prácticas en muchos aspectos previamente evaluados e incluso una cultura de la comunicación muy alta, la confianza entre los miembros es parcial, por esto tanto D, C y E fueron intervenidas con actividades de refuerzo de confianza entre miembros, y en dichas actividades se logró evidenciar que incluso algunos miembros no conocían a sus colegas en los equipos de trabajo (en un nivel profesional). A por otro lado demostró que actividades de comunicación exhaustivas no eran requeridas para confiar en sus colegas, mostrando un trabajo más independiente pero igualmente sinérgico.

Como actividades para continuar nutriendo aspectos como el empoderamiento de los miembros de los equipos, la confianza y la comunicación, se planteó la implementación de reuniones de inicio y fin de jornada (la implementación puede ser variable, sólo inicio, sólo fin, ambas) con una duración corta comprendida entre los 15 y 20 minutos con el fin de:

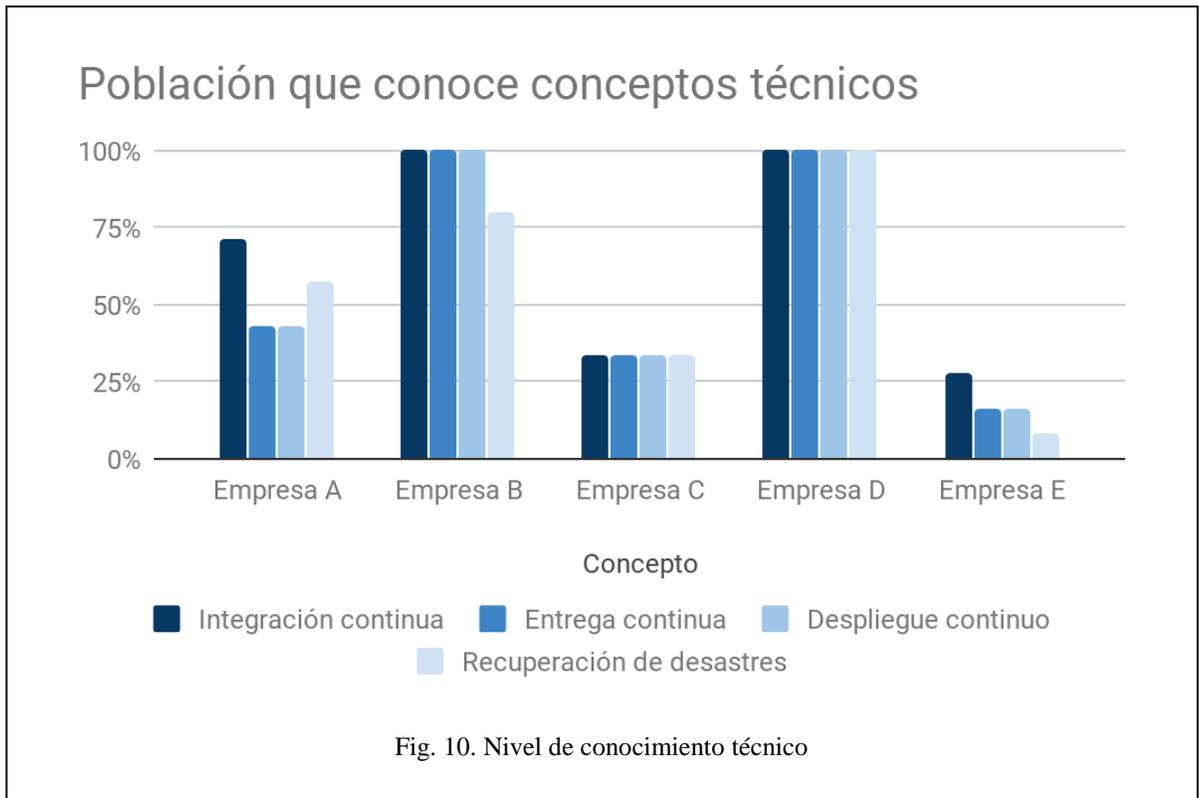
- Socializar el estado de las actividades asignadas a cada miembro.
- Informar a todos los miembros del equipo cuál sería el objetivo del día que cada uno se proponía.
- Aportar con consejos o experiencias previas de otros miembros a los miembros que afrontan algún reto previamente solucionado por otro.
- Compartir experiencias aprendidas de la jornada para nutrir el registro.

Al final de la etapa los mismos miembros de los equipos lograron expresar que la sinergia mejoró, pues conocer las necesidades y actividades de otros miembros les

impulsaba a buscar una solución común favoreciendo no sólo la adquisición de objetivos personales.

5) Conocimientos técnicos asociados a DevOps

En cuanto al pilar de prácticas tecnológicas entre los resultados obtenidos se evaluaron los conocimientos de los conceptos asociados a DevOps, este primer apartado no significa que dichos conceptos se están aplicando dentro de la empresa, pero partir del conocimiento de ellos facilita su posterior implementación de no ser existente. A continuación, en la figura 10, se muestran los resultados obtenidos:



También dentro de la información recolectada se tiene el conjunto de prácticas que actualmente las empresas implementan, con ello se realizó la revisión en contraste de los conocimientos de los miembros vs la implementación, en cuanto a las prácticas, estas se listan a continuación:

TABLA 5. PRACTICAS TECNOLÓGICAS POR EMPRESA

Prácticas del pilar tecnológico asociadas a DevOps					
Práctica	Empresa A	Empresa B	Empresa C	Empresa D	Empresa E
Uso de SCM	X	X	X	X	
Gestión de dependencias	X	X	X	X	X
Automatización de procesos		X			
Automatización de pruebas		X		X	
Automatización de auditoría				X	
Planes de recuperación a desastres				X	
Virtualización	X	X		X	X

Ahora y concluyendo con base a los resultados anteriores, según la tabla 5, el comportamiento de la implementación de las prácticas en contraste de los conocimientos de los miembros de estas es compatible, es decir, las prácticas que no figuran en implementación son las que agrupan los conceptos que son desconocidos por los miembros. Para este punto nuevamente B y D muestran unos resultados muy favorecedores tanto en el marco conceptual como en las prácticas implementadas, pero destaca D, con quien sólo fue necesario planificar un plan de capacitación en automatización de procesos, mientras B requirió apoyo en los procesos de automatización de auditorías y los planes de recuperación a desastres. Un resultado que no se esperaba fue la ausencia de implementación de procesos que A muestra, a pesar de cumplir con facilidad en muchos apartados anteriores, fue evidente la ausencia de procesos y esto apunta directamente a la falta del marco conceptual dentro de los miembros, de manera tal que A, C y E requirieron prácticamente un plan de capacitación igual, además de un refuerzo de buenas prácticas en desarrollo de software; se invitó a miembros de los equipos a participar de actividades de formación profesional desarrolladas por las mismas empresas con apoyo de los autores de este trabajo de investigación, que incluían el diplomado “Entorno de aplicaciones web,

full stack web developer” impartido por uno de los autores de este mismo trabajo y patrocinado por el MinTIC de Colombia mediante su plan FormATIC.

6) Drivers de DevOps

Se consultó a las empresas por indicadores que podrían medir el desempeño con respecto a drivers clave para DevOps, entre la información recolectada se obtuvo la siguiente.

a. Tiempo de respuesta ante situaciones críticas

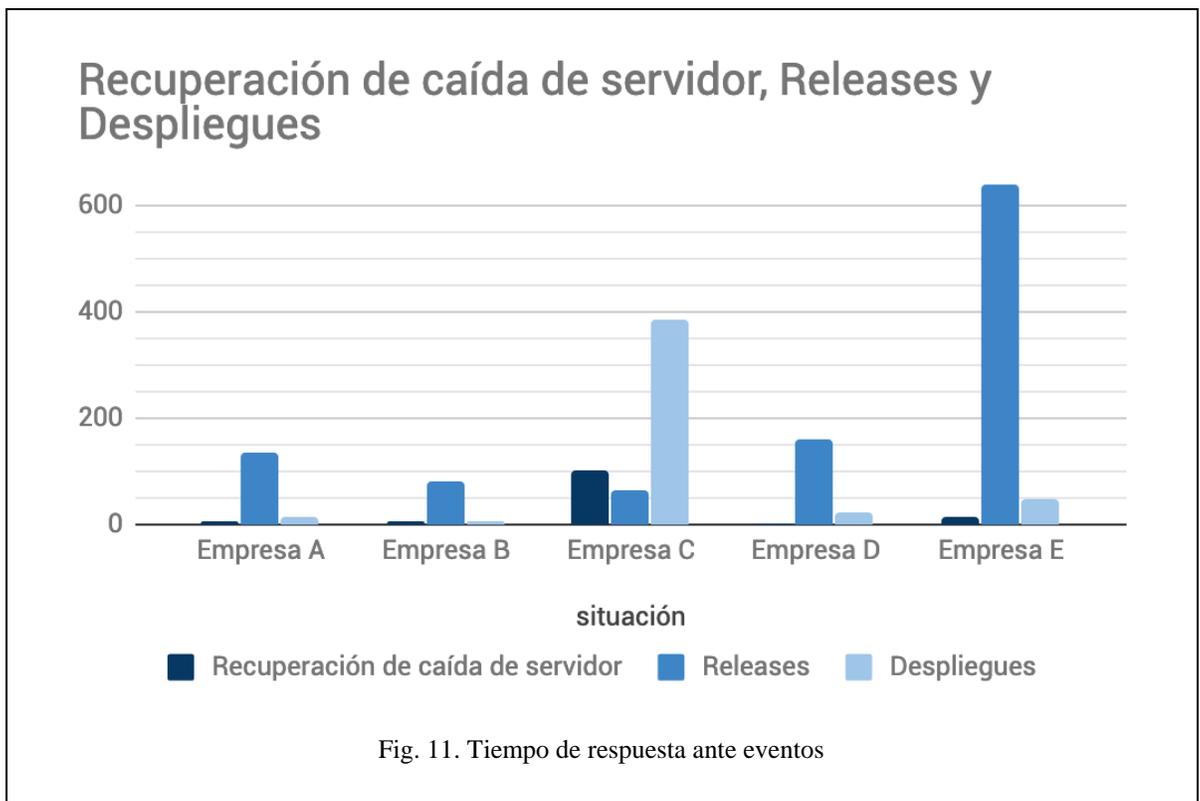
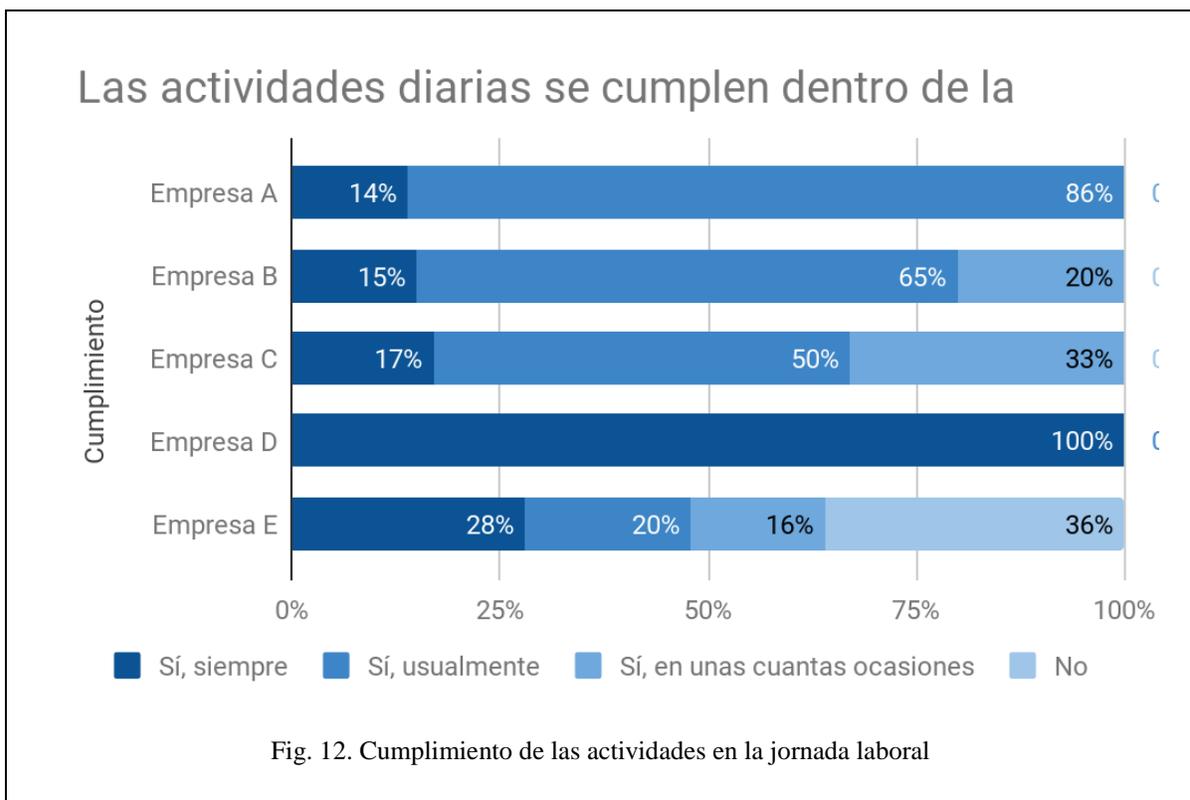


Fig. 11. Tiempo de respuesta ante eventos

Para la interpretación de este gráfico, fig. 11, entre menor sea el tamaño de las barras, mejor es el tiempo de respuesta las situaciones planteadas, así mismo implicó una menor intervención en dichos procesos en las empresas, siendo crítica la situación actual de C y E.

b. Tiempo de respuesta ante situaciones críticas

La satisfacción del cliente es tan importante como la percepción misma de los miembros de la empresa con ella misma, por eso es importante garantizar espacios de trabajo a menos que permitan el cumplimiento de sus actividades durante los espacios establecidos y con ello también los tiempos para la cualificación del personal, pero esto último no suele existir debido a la falta de espacios porque las actividades toman incluso tiempo extracurricular, como se puede ver en la fig. 12.

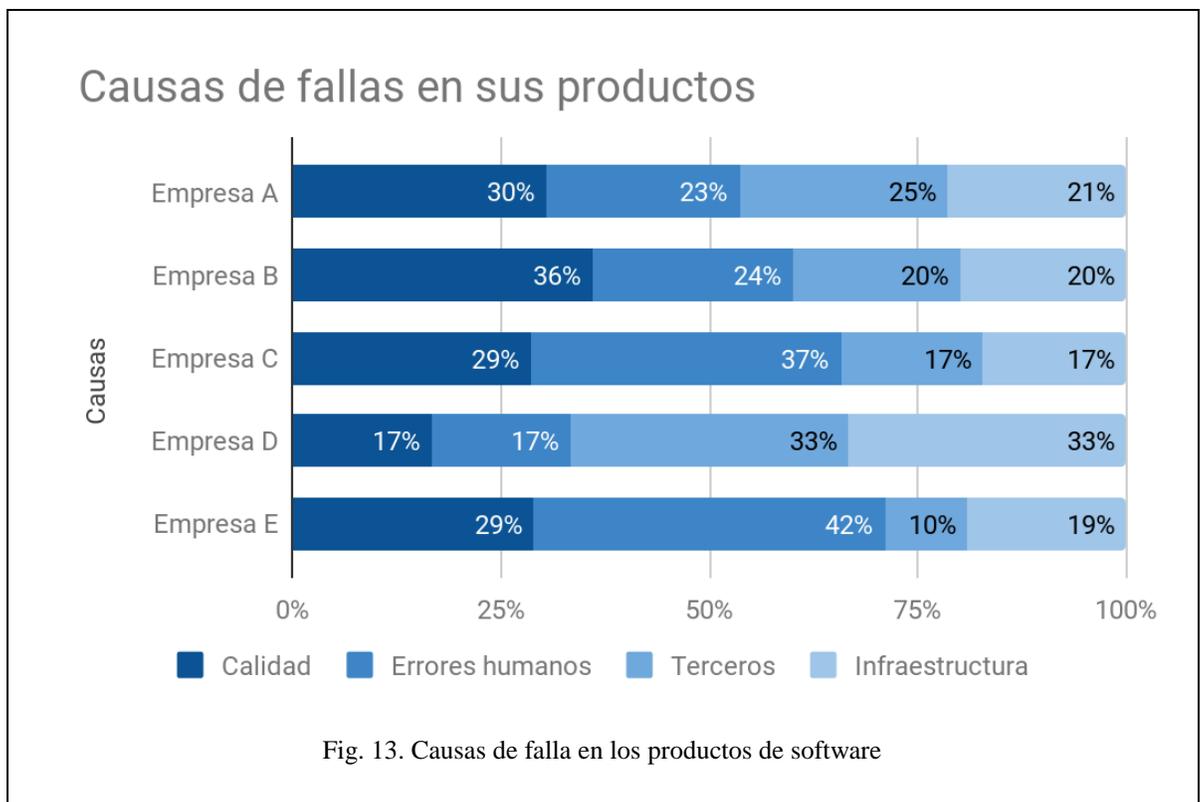


El estado ideal de este factor sería que las actividades siempre pudiesen ser concretadas durante el tiempo establecido para las mismas y en esto interfieren tanto los posibles reprocesos por mala elicitación de requerimientos, como los errores humanos, entre otros factores; acorde a los resultados obtenidos, fig. 12, sólo D garantiza a sus miembros el tiempo adecuado para sus labores

respetando las jornadas pautadas, mientras que A, B y C aunque presentan contratiempos, son aceptables y se consideró la oportunidad de mejora para ambas empresas revisando los factores que originan el problema. D por su lado tuvo que ser sometida a una revisión de los factores causantes del resultado tan desalentador que obtuvo, que posteriormente estuvo muy relacionado con las metodologías y la falta de comunicación que hacían de los errores un proceso costoso tanto en tiempo como en recursos monetarios.

c. Causas de fallas en productos de software

Entre la información recolectada se solicitó a las empresas valorar la incidencia de algunas causas identificadas como causas comunes en fallas de los productos de software dentro de la industria del desarrollo de software (enfocado en Colombia), los resultados fueron los siguientes, fig. 13:



Posicionando como causas más incidentes a la calidad y los errores humanos, ver fig. 13, siendo ambas tratadas de manera implícita al adoptar adecuadamente DevOps. Con esta información entonces se planteó darles prioridad a los temas automatización de procesos para contrarrestar los errores humanos y apoyar a la mejora de la calidad del producto, así mismo a los procesos de auditoría y pruebas automatizadas.

7) Cierre de la etapa 1

Una vez las empresas lograron posicionarse en el cuadrante adecuado según los objetivos de su implementación previamente definidos, se da cierre a la primera etapa para iniciar el trabajo de la siguiente; como resultados del trabajo realizado, todos los miembros de la empresa participantes de esta investigación conocían y contaban con las competencias blandas (comunicación, colaboración y adaptabilidad) y las competencias técnicas requeridas para afrontar el reto que el framework plantea, además la cultura de las organizaciones se encontraba lista para asumir retos como el empoderamiento de los empleados, la investigación y la innovación.

Durante esta etapa dos de las empresas se retiran del estudio (A y D) debido a la poca disponibilidad de tiempos, no sin expresar que procurarán retomar la estrategia en un momento más cómodo para ellas; dentro de su percepción las juntas directivas y los mismos miembros de los equipos se encontraron positivamente impactados.

C. ETAPA 2. DIAGNÓSTICO DE COMPATIBILIDAD CON LA TECNOLOGÍA, PRÁCTICAS TECNOLÓGICAS E INFRAESTRUCTURA ACTUAL

Concluida la etapa anterior el enfoque se cambió a revisar las tecnologías y prácticas tecnológicas de las empresas y su compatibilidad con los conceptos previamente aprendidos acordes a DevOps, una vez la revisión se llevó a cabo se diligenciaron los resultados agrupados por las categorías (1) tecnología, (2) frameworks y (3) servidores, la infraestructura es un caso muy puntual y varió en cada empresa por su enfoque pues algunas en lugar de ofrecer servicios SaaS, lo desarrollan para terceros y no tienen repercusión en estos temas más que para servidores de desarrollo que simulen los servidores de producción de sus clientes.

Se planteó la implementación de Líneas Automatizadas de Producción de Software dentro de las empresas con el fin de cubrir el conjunto prácticas asociadas al pilar de tecnología e infraestructura. A continuación, en la figura 14, se presenta el planteamiento de estas:

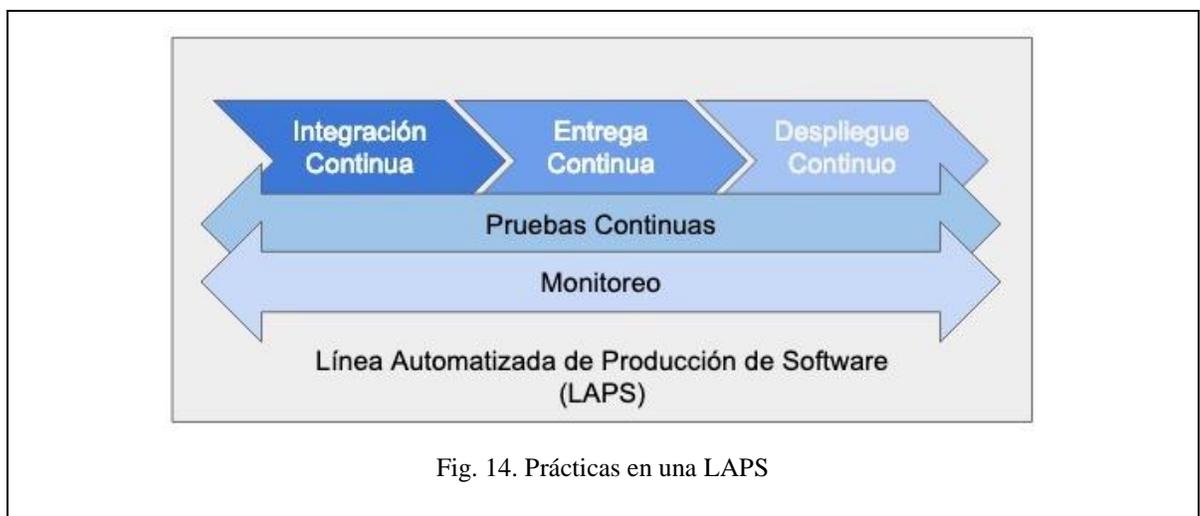


Fig. 14. Prácticas en una LAPS

Dichas líneas consideraron la integración tanto con infraestructuras totalmente en la nube como híbridas.

A continuación, en la figura 15, se muestran los resultados de la consulta de tecnologías identificadas en las empresas, su compatibilidad con aspectos críticos de la adopción y su estado de compatibilidad.

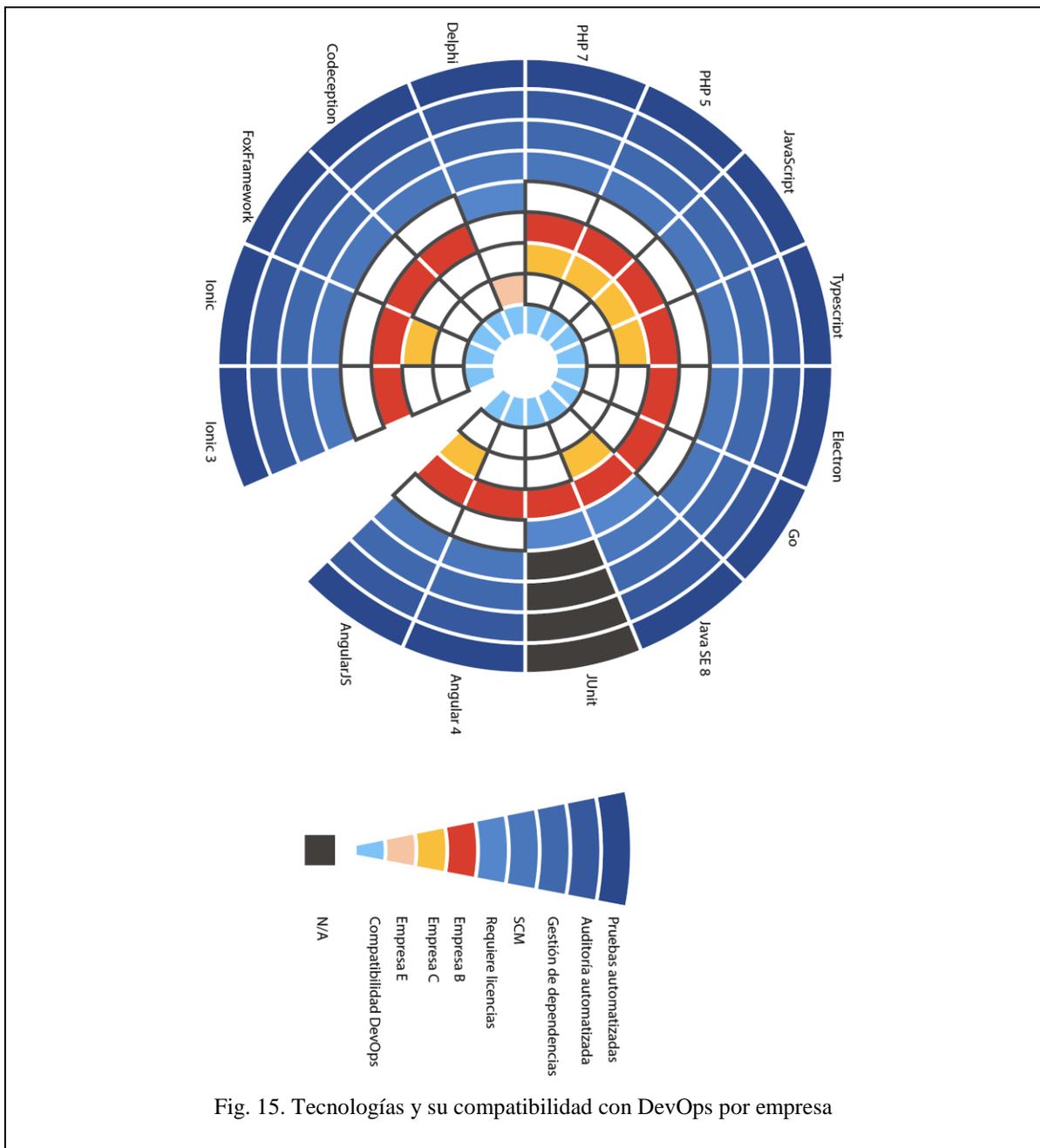


Fig. 15. Tecnologías y su compatibilidad con DevOps por empresa

Para el caso de la empresa E, su tecnología requería del pago de licencias para adicionar la funcionalidad de pruebas automatizadas y auditoría, dicho presupuesto no estaba contemplado en los planes de inversión, obstaculizando el desarrollo de la LAPS y con ello la capacidad de cumplir con el pilar y por ésta razón el proceso de adopción de DevOps fue suspendido en este punto. Mientras tanto los procesos con C permitieron el desarrollo de la LAPS, pero posterior a las capacitaciones el personal que inicialmente fue capacitado como líderes de área y quien sería el responsable de la demás capacitación interna a sus homólogos dejó la empresa por nuevas oportunidades laborales en empresas muchos más grandes que consideraron sus perfiles muy atractivos, esto se atribuyó en gran parte a la formación recibida y la necesidad del mercado de personal con experiencia en el framework; esto deja solamente a B como candidato para realizar la validación de la implementación en la siguiente etapa de la estrategia.

D. ETAPA 3. VALIDACIÓN DE LA IMPLEMENTACIÓN DE LA CULTURA ORGANIZACIONAL, PRÁCTICAS TECNOLÓGICAS E INFRAESTRUCTURA REQUERIDA

Posterior a la implementación de las prácticas tanto culturales como tecnológicas y de infraestructura dentro de la empresa B (y puntualmente un proyecto de desarrollo), los resultados obtenidos después de la implementación de la estrategia para la adopción de DevOps se exponen a continuación:

TABLA 6. RESULTADOS DE LA IMPLEMENTACIÓN DEVOPS EN LA EMPRESA B

	Media Histórica (meses)	Resultado obtenido (meses)	Mejora
Tiempo de desarrollo hasta MVP	6	2	70%

	Software Previo	Software Actual	Mejora
Percepción positiva de los usuarios	35%	95%	60%

Nota: datos recolectados de los estados MVP de cada producto

	Media Histórica (horas)	Media actual obtenido (horas)	Mejora
Tiempos de respuesta/mantenimiento	72	4	90%

	Media histórica (mensual)	Media actual (mensual)	Mejora
Disponibilidad del producto	83,33%	99,72%	16,39%

	Media histórica (errores/release)	Media actual (errores/release)	Mejora
Errores detectados en producción	37	2	95%

	Media histórica (errores/release)	Media actual (errores/release)	Mejora
Errores detectados en desarrollo	No existen registros	14	Sin registros suficientes

	Media histórica	Media actual
Prevención de errores en producción	Sin registros suficientes	86%

	Media histórica (meses)	Media actual (mes)	Mejora
Despliegues	0,25	6	2400%

	Media histórica	Media actual
Despliegues exitosos	No existen registros	100%

Los datos para la “media actual” fueron recolectados durante 3 meses.

Aunque los resultados sobre los criterios a monitorear establecidos para la revisión del impacto de la implementación en el proceso de la empresa B fueron muy positivos, como se muestra en la tabla 6, se espera que a medida que sigan el proceso de mejora continua estos sean aún mejores con el tiempo, en todo caso, los resultados obtenidos son satisfactorios, destacando puntualmente que aunque no existen registros previos de los despliegues exitosos, la empresa no lograba un margen superior al 80% de éxito en esta práctica, esto según la entrevista con los equipos, debido principalmente a errores de comunicación entre desarrollo y operaciones, seguido de errores humanos, ahora junto con el plan de monitoreo y mejora continua se llevan los registros de los despliegues exitosos (junto con todos los criterios previamente mencionados) y su tasa del 100% de despliegues exitosos cumple con la primera métrica a evaluar de este trabajo de investigación (Deployment Success Rate), mientras que en términos del Application Error Rate se obtiene un 0,23% (usando la fórmula $AER = 100 * \text{Errores} / \text{Puntos de prueba}$),

comparado con la anterior tasa de 4,38% la reducción de errores en producción por los procesos implicados en la LAPS implementada asciende a un 95%, además de obtener una tasa de prevención de errores desde desarrollo a producción del 86%, todo esto enriqueciendo la calidad del producto de software e incluso con mejoras en tiempos de desarrollo de un 70% y la capacidad de tiempos de respuesta 90% más rápidos y el incremento de despliegues en un 2400%. Quizá uno de los factores más gratificantes fue el resultado de la mejora de la percepción del producto por el cliente, la cual incrementó en un 60% para un total de 95% de percepción positiva.

V. CONCLUSIONES Y TRABAJOS FUTUROS

E. CONCLUSIONES

- Con base a los resultados obtenidos en la investigación, puntualmente en la etapa 3 con respecto al Deployment Success Rate y Application Error Rate, la estrategia diseñada e implementada en la empresa B, logra aportar a la calidad del producto de software de la MIPYMES colombiana desarrolladora de software, entre otros aspectos culturales, de experiencia del cliente y del negocio.
- La división por etapas de la estrategia permite a las empresas trabajar una transición progresiva que permite la preevaluación del impacto en recursos necesarios para cada una según los resultados de cada diagnóstico.
- El pensamiento de desarrollo de software tradicional (metodologías tradicionales, división por equipos con alta hermeticidad, entre otros factores) presenta resistencia a la implementación de DevOps.
- El uso de tecnologías propietarias o la tercerización de procesos/departamentos dificultan la implementación del framework por la falta de control de estas, el costo de crecimiento y/o el no alineamiento de los objetivos empresariales.

F. TRABAJOS FUTUROS

Como continuación a este trabajo se plantea:

El desarrollo de los demás drivers planteados para DevOps, diseñando estrategias para atacar los retos que cada uno de ellos plantea, para posteriormente evaluar el impacto con las métricas planteadas para cada uno de ellos.

El desarrollo de mecánicas que permitan a las MIPYMES que implementan DevOps, combatir la deserción de los miembros capacitados a empresas más grandes, impactando enormemente la conservación del conocimiento en las empresas, tornándose así en semilleros de formación profesional usados para catapultarse a empresas más grandes.

REFERENCIAS

- [1] J. Wettinger, U. Breitenbucher, and F. Leymann, "Standards-based DevOps automation and integration using TOSCA," 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, Dec. 2014.
- [2] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "Dimensions of DevOps," in *Agile Processes in Software Engineering and Extreme Programming*. Springer Science + Business Media, 2015, pp. 212–217.
- [3] N. Kerzazi and B. Adams, "Who needs release and devops engineers, and why?," *Proceedings of the International Workshop on Continuous Software Evolution and Delivery - CSED '16*, 2016.
- [4] M. Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery," *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)*, May 2015.
- [5] Z. Babar, A. Lapouchnian, and E. Yu, "Modeling DevOps deployment choices using process architecture design dimensions," in *Lecture Notes in Business Information Processing*. Springer Science + Business Media, 2015, pp. 322–337.
- [6] G. V. Hulme, "Webinars & hangouts," in *Features*, DevOps.com, 2015. [Online]. Available: <http://devops.com/2015/03/18/jeff-sussna-on-design-thinking-and-devops/>. Accessed: Sep. 15, 2016.
- [7] Puppet, "2016 state of DevOps report," Puppet, 2016. [Online]. Available: <https://puppet.com/resources/white-paper/2016-state-of-devops-report>. Accessed: Sep. 12, 2016.
- [8] R. Jabbari, N. Ali, K. Petersen, B. Tanveer, "What's DevOps? A Systematic Mapping Study on Definitions and Practices", *XP 16' Workshops*, ACM, 2016.
- [9] proexport Colombia. "Industria de tecnologías de información. Technical report", 2008.
- [10] S. Martínez Marín, "Evaluación de Estrategias de Crecimiento en la Industria del Software en Colombia con Dinámica de Sistemas", *Maestría en Ingeniería de Sistemas*, Universidad Nacional de Colombia, 2013.

- [11] S. Martínez Marín, “El crecimiento de la industria del software en Colombia: un análisis sistémico”, Escuela de Ingeniería Antioquia, EIA Vol. 12 Ed. 23, pp.95-106, Evigado, Colombia, 2015.
- [12] New Relic, “DevOps without measurement is a fail”, New Relic, 2015.
- [13] Ayuntamiento de Barcelona, “Test Innova”. [Online]. Available: http://www.bcn.cat/movies/innova/test_innova/index-es.html. Accessed: May. 1, 2017.
- [14] K. Nybom, J. Smeds, and I. Porres, "On the impact of mixing responsibilities between Devs and Ops," in Agile Processes, in Software Engineering, and Extreme Programming. Springer Nature, 2016, pp. 131–143.
- [15] Sonatype, “DEVSECOPS COMMUNITY SURVEY 2017”, 2017.
- [16] E. Woods, "Operational: The forgotten architectural view," IEEE Software, vol. 33, no. 3, pp. 20–23, May 2016.
- [17] D. Spinellis, "Don't install software by hand," IEEE Software, vol. 29, no. 4, pp. 86–87, Jul. 2012.
- [18] B. Fitzgerald and K.-J. Stol, "Continuous software engineering and beyond: Trends and challenges," Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014, 2014.
- [19] L. E. Lwakatare et al., "Towards DevOps in the embedded systems domain: Why is it so hard?," 2016 49th Hawaii International Conference on System Sciences (HICSS), Jan. 2016.
- [20] Federación Colombiana de la Industria de Software, "Informe de caracterización del sector de software y tecnologías de la información en Colombia", 2015.
- [21] Federación Colombiana de la Industria de Software. “Descripción del sector del software. Technical report”, 2008.
- [22] PractiTest. “Stateof testing Report 2018”, 2018.
- [23] T. Hynninen, J. Kasurinen, A. Knutas, O. Taipale, “Software Testing: Survey of the Industry Practices”, MIPRO 2018, May 21-25, 2018, Opatija Croatia, 2018.
- [24] G. Kim, P. Debois, J. Willis, J. Humble and J. Allspaw, The DevOps handbook.
- [25] J. Lee, S. Kang and D. Lee, "Survey on software testing practices", IET Software, vol. 6, no. 3, p. 275, 2012. Available: 10.1049/iet-sen.2011.0066.

- [26] B. Paul, "Is your Cloud Elastic Enough? Performance Modeling the Elasticity of Infrastructure as a Service (IaaS) Cloud Applications", *Proceeding, ICPE '12 Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, Pages 263-266, 2012.
- [27] S. Douglas, W. Jules, G. Christopher, "Elastic Infrastructure to Support Computing Clouds for Large-Scale Cyber-Physical Systems", *2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, 2014.
- [28] C. Michael, T. Brendan, B. Theophilus, V Laurent, "Chaos Monkey: Increasing SDN Reliability through Systematic Network Destruction", *CCR*, 2015.
- [29] "Principles of Chaos Engineering", *Principlesofchaos.org*, 2019. [Online]. Available: <http://principlesofchaos.org/>. [Accedido: 14- Mar- 2018].
- [30] "Enterprise Application Container Platform | Docker", *Docker*, 2019. [Online]. Available: <https://www.docker.com/>. [Accedido: 14- Mar- 2018].
- [31] O. White, K. B. M. Manager, and J. KabanovCEO, "DevOps productivity report 2013," *ZeroTurnaround*, 2013.
- [32] H. B. Christensen, "Teaching DevOps and cloud computing using a cognitive apprenticeship and story-telling approach," *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '16*, 2016.
- [33] L. Zhu, L. Bass, and G. Champlin-Scharff, "DevOps and its practices," *IEEE Software*, vol. 33, no. 3, pp. 32–34, May 2016.
- [34] J. Smeds, K. Nybom, and I. Porres, "DevOps: A definition and perceived adoption impediments," *Agile Processes, in Software Engineering, and Extreme Programming*, pp. 166–177, 2015.
- [35] L. Bass, I. M. Weber, and L. Zhu, *DevOps: A software architect's perspective*, 1st ed. Addison-Wesley, 2015, ch. 1. What Is DevOps?
- [36] A. Wahaballa, O. Wahballa, M. Abdellatief, H. Xiong, and Z. Qin, "Toward unified DevOps model," *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Sep. 2015.
- [37] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE Software*, vol. 33, no. 3, pp. 94–100, May 2016.
- [38] C. A. Cois, J. Yankel, and A. Connell, "Modern DevOps: Optimizing software development through effective system interactions," *2014 IEEE International Professional*

Communication Conference (IPCC), Oct. 2014.

[39] M. Callanan and A. Spillane, "DevOps: Making it easy to do the right thing," *IEEE Software*, vol. 33, no. 3, pp. 53–59, May 2016.

[40] J. Wettinger, V. Andrikopoulos, and F. Leymann, "Enabling DevOps collaboration and continuous delivery using diverse application environments," in *On the Move to Meaningful Internet Systems: OTM 2015 Conferences*. Springer Science + Business Media, 2015, pp. 348–358.

[41] J. Wettinger, U. Breitenbücher, and F. Leymann, "Compensation and Convergence — comparing and combining deployment automation approaches," *International Journal of Cooperative Information Systems*, vol. 24, no. 03, p. 1541001, Sep. 2015.

[42] J. Wettinger, U. Breitenbücher, O. Kopp, and F. Leymann, "Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel," *Future Generation Computer Systems*, vol. 56, pp. 317–332, Mar. 2016.

[43] J. Wettinger, V. Andrikopoulos, and F. Leymann, "Automated capturing and systematic usage of DevOps knowledge for cloud applications," *2015 IEEE International Conference on Cloud Engineering*, Mar. 2015.

[44] S. Hosono, "A DevOps framework to shorten delivery time for cloud applications," *International Journal of Computational Science and Engineering*, vol. 7, no. 4, p. 329, 2012.

[45] I. K. Nybom, J. Smeds, and I. Porres, "On the impact of mixing responsibilities between Devs and Ops," in *Agile Processes, in Software Engineering, and Extreme Programming*. Springer Nature, 2016, pp. 131–143.

[46] I. Centro de Innovación Pública Digital, "Infografía: cultura de innovación", [Online]. Available: Vive Digital, <http://centrodeinnovacion.gobiernoonline.gov.co/es/caja-herramientas/infografia-cultura-de-innovacion>.

[47] Puppet. "DevOps Assessment", [Online]. Available: <https://puppet.com/devops-assessment>.

[48] P. Bejarano, A. Heredia, H. Ordoñez, "Lineamientos para un acercamiento a la implementación de DevOps para MIPYMES en Colombia," in *Obras colectivas en ciencias de la computación*. 2018, pp. 505–511. ISBN 978-958-5415-19-5.