



**FACULTAD DE POSTGRADO**

**TESIS DE POSTGRADO**

**PROPUESTA DE DISEÑO DE UN MODELO DE INTEGRACIÓN DE  
DEVOPS Y METODOLOGÍAS ÁGILES CON ENFOQUE ITIL V4 EN  
LA CONSULTORA NGS**

**SUSTENTADO POR:**

**MAZIE ELIZABETH LAINEZ REYES  
FERNANDO DANIEL LÓPEZ BARAHONA**

**PREVIA INVESTIDURA AL TÍTULO DE**

**MÁSTER EN  
GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN**

**TEGUCIGALPA, FRANCISCO MORAZÁN, HONDURAS, C.A.**

**FEBRERO, 2026**

**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA  
UNITEC**

**FACULTAD DE POSTGRADO**

**AUTORIDADES UNIVERSITARIAS**

**RECTORA**

**ROSALPINA RODRÍGUEZ**

**VICERRECTOR ACADÉMICO NACIONAL**

**JAVIER ABRAHAM SALGADO LEZAMA**

**DECANA FACULTAD DE POSTGRADO**

**ANA DEL CARMEN RETTALLY VARGAS**

**PROPUESTA DE DISEÑO DE UN MODELO DE INTEGRACIÓN DE  
DEVOPS Y METODOLOGÍAS ÁGILES CON ENFOQUE ITIL V4 EN  
LA CONSULTORA NGS**

**TRABAJO PRESENTADO EN CUMPLIMIENTO DE LOS  
REQUISITOS EXIGIDOS PARA OPTAR AL TÍTULO DE  
MÁSTER EN  
GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN**

**ASESOR METODOLÓGICO**

**JORGE RAÚL MARADIAGA CHIRINOS**

**MIEMBROS DE LA TERNA:**

**ANTHONY STEVE BARAHONA ESPINOZA  
ELVIN OSMAN BOBADILLA SALINAS  
KEVIN EDUARDO FUNEZ FUNEZ**



**FACULTAD DE POSTGRADO**  
**PROPUESTA DE DISEÑO DE UN MODELO DE INTEGRACIÓN DE**  
**DEVOPS Y METODOLOGÍAS ÁGILES CON ENFOQUE ITIL V4 EN**  
**LA CONSULTORA NGS**

**Mazie Elizabeth Lainez Reyes**  
**Fernando Daniel López Barahona**

**Resumen**

El presente trabajo desarrolla una propuesta de integración de prácticas DevOps y metodologías ágiles con enfoque ITIL v4 para fortalecer el flujo de integración y entrega continua (CI/CD) en la consultora tecnológica Nueva Generación de Software (NGS). El estudio parte del reconocimiento de brechas operativas asociadas a automatización limitada, deficiencias en trazabilidad, ausencia de lineamientos técnicos estandarizados y gestión del servicio insuficiente, factores que afectan la confiabilidad y eficiencia de los despliegues de software. La investigación se desarrolló bajo un enfoque aplicado, con diseño no experimental y alcance descriptivo-propositivo, empleando análisis documental y diagnóstico interno para caracterizar el estado actual del flujo CI/CD y los procesos relacionados. Los resultados permitieron identificar prácticas críticas susceptibles de mejora y establecer una base para el diseño de la propuesta. A partir del marco teórico de DevOps, Scrum e ITIL v4, se formuló un modelo integrado orientado a fortalecer la automatización, trazabilidad, control y gestión del servicio durante el ciclo de entrega. La propuesta incluye un plan estratégico de cierre de brechas, lineamientos técnicos y operativos, plan de formación, medidas de control e indicadores basados en métricas de desempeño CI/CD, así como un cronograma estructurado por fases para su implementación en NGS. Se espera que la adopción del modelo contribuya a mejorar la estabilidad de los despliegues, reducir tiempos de entrega, fortalecer la gestión del servicio y consolidar una cultura de mejora continua, generando impacto en la eficiencia operativa y en la calidad del software entregado al cliente.

- **Palabras clave:** Automatización del flujo de software, CI/CD, DevOps, Gestión del servicio, Integración de prácticas, ITIL v4, Mejora continua, Scrum.



**FACULTAD DE POSTGRADO  
PROPUESTA DE DISEÑO DE UN MODELO DE INTEGRACIÓN  
DE DEVOPS Y METODOLOGÍAS ÁGILES CON ENFOQUE  
ITIL V4 EN LA CONSULTORA NGS**

**Mazie Elizabeth Lainez Reyes  
Fernando Daniel López Barahona**

**Abstract**

This work proposes integrating DevOps practices and agile methodologies with an ITIL v4 focus to strengthen the continuous integration and delivery (CI/CD) pipeline at the consulting firm Nueva Generación de Software (NGS). The study identifies operational gaps related to limited automation, insufficient traceability, lack of standardized technical guidelines, and weak service management factors that affect reliability and efficiency in software deployments. The research follows an applied, non-experimental, descriptive–propositional design, using document analysis and internal diagnostics to characterize the current CI/CD pipeline and related processes. The findings highlight critical practices requiring improvement and provide the basis for the proposed model. Drawing on DevOps, Scrum, and ITIL v4, an integrated model is formulated to strengthen automation, traceability, control, and service management across the delivery lifecycle. The proposal includes a strategic gap-closing plan, technical and operational guidelines, a training plan, control measures, CI/CD performance indicators, and a phased implementation schedule. Adoption of this model is expected to improve deployment stability, reduce delivery time, strengthen service management, and consolidate a culture of continuous improvement, positively impacting operational efficiency and software quality delivered to the client.

**Keywords:** CI/CD, Continuous improvement, DevOps, ITIL v4, Practice integration, Scrum, Service management, Software workflow automation.

## DEDICATORIA

### – Mazie Elizabeth Lainez Reyes

Dedico mi trabajo final, en primer lugar, a Dios, por haberme concedido la salud, la sabiduría y la perseverancia en cada paso de este proceso de aprendizaje, para culminar este importante logro académico. Su guía constante ha sido fundamental en cada etapa de este camino.

A mi esposo, por su amor, apoyo incondicional y comprensión, por acompañarme con paciencia en los momentos de esfuerzo y sacrificio, y por creer en mí incluso cuando el camino parecía complicado.

A mis hijos, quienes son mi mayor inspiración y la razón principal para seguir creciendo personal y profesionalmente. Este logro es también para ustedes, con la esperanza de ser un ejemplo de que la constancia, la disciplina y la fe permiten alcanzar los sueños.

A mis padres, por su amor, sacrificio y valores inculcados desde siempre. Gracias por su apoyo constante, por motivarme a superarme y por ser el pilar fundamental de mi formación personal y profesional.

A todos ellos, dedico este logro con profundo agradecimiento y amor.

### – Fernando Daniel López Barahona:

A **Dios**, por la fortaleza y sabiduría que me ha concedido en cada etapa de mi vida.

A mi **madre**, por ser el pilar de nuestra familia, por sus sacrificios incansables, su ejemplo de esfuerzo y su apoyo incondicional para que sus hijos sigamos creciendo y avanzando.

A mi **abuela** materna, quien me crio con amor y disciplina, y me transmitió valores morales que han guiado mi formación. Sus sacrificios y enseñanzas han sido fundamentales para convertirme en una persona de bien y un profesional digno.

A mi **familia**, por acompañarme en este camino y motivarme a ser cada día mejor. A mis **hermanos** menores, con la esperanza de que este logro sea un recordatorio de que los sueños se alcanzan con dedicación y perseverancia.

A mi **padre**, porque aunque no convivimos juntos, me ha demostrado a través de su ejemplo que el esfuerzo y el trabajo constante permiten alcanzar los objetivos que nos proponemos.

Con gratitud y cariño, les dedico este logro.

## AGRADECIMIENTO

– **Mazie Elizabeth Lainez Reyes**

Agradezco en primer lugar a **Dios**, por su amor, misericordia y guía constante, por darme la fortaleza, la sabiduría y la paz necesarias para culminar este trabajo académico. Cada paso de este camino ha sido posible gracias a Su bendición.

Quiero expresar mi más sincero agradecimiento a mi compañero de tesis **Fernando López**, cuya dedicación, esfuerzo y compromiso fueron fundamentales para la realización de este trabajo. Su apoyo constante, sus aportes académicos y su disposición para enfrentar juntos los retos que se presentaron hicieron de este proceso una experiencia enriquecedora y de gran aprendizaje. Más allá de lo académico, valoro la colaboración, la paciencia y la motivación que compartimos, que sin duda marcaron una diferencia en la culminación exitosa de nuestra investigación.

Deseo expresar mi más profundo agradecimiento a la empresa **Nueva Generación de Software**, en la que laboré, por haberme brindado el apoyo, los recursos y la confianza necesarios para llevar a cabo esta investigación. La oportunidad de aplicar mis conocimientos en un entorno profesional, junto con el acompañamiento de mis colegas y superiores, fue fundamental para enriquecer este trabajo. Reconozco el valor de las experiencias adquiridas y el aprendizaje constante que me ofrecieron, los cuales contribuyeron de manera significativa al desarrollo y culminación de esta tesis.

Agradecimiento a mis compañeros **Julio Flores y Meylin Blanco**, quienes fueron un pilar fundamental durante este proceso de la maestría. Su apoyo constante, la colaboración y la amistad que compartimos hicieron que cada reto fuera más llevadero y cada logro más significativo. Más allá del trabajo académico, valoro profundamente la confianza, el ánimo y la perseverancia que nos unieron, convirtiendo esta experiencia en un camino enriquecedor tanto en lo profesional como en lo personal.

Mi agradecimiento al Msc. **Jorge Raúl Maradiaga Chirinos**, por su acompañamiento, apoyo y valiosas enseñanzas, que guiaron este trabajo con orden, claridad y rigor académico, que Dios bendiga abundantemente su vida y le dé el mayor de los éxitos en todas las labores que desempeña con excelencia.

Mil gracias a cada una de las personas que contribuyeron a la culminación de este logro.

– **Fernando Daniel López Barahona:**

Agradezco a **Dios** por la sabiduría, fortaleza y claridad que me ha concedido a lo largo de este camino académico y profesional.

Extiendo mi agradecimiento a **Nueva Generación de Software** por el apoyo brindado durante el desarrollo de este trabajo y por permitirme aplicar los conocimientos adquiridos en un entorno real.

A mis compañeros de maestría y amigos **Julio Flores** y **Meylin Blanco**, por su colaboración, compromiso y disposición durante las actividades realizadas.

A mi compañera de tesis, **Mazie Laínez**, por su esfuerzo, dedicación y acompañamiento constante durante estos últimos seis meses, en los que trabajamos en conjunto para alcanzar los objetivos planteados.

Agradezco también al **Msc. Jorge Raúl Maradiaga Chirinos**, asesor metodológico, por guiarnos con profesionalismo, paciencia y orientación académica en las fases claves de nuestro Trabajo Final de Graduación, y por mostrarnos el camino para desarrollar una investigación rigurosa y con sentido práctico.

A todos quienes aportaron tiempo, conocimiento o palabras de aliento, reciban mi sincero agradecimiento.

# ÍNDICE DE CONTENIDO

<b>DEDICATORIA</b> .....	<b>xi</b>
<b>AGRADECIMIENTO</b> .....	<b>xii</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>xix</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>xxii</b>
<b>CAPÍTULO I - PLANTEAMIENTO DE LA INVESTIGACIÓN</b> .....	<b>1</b>
<b>1.1 INTRODUCCIÓN</b> .....	<b>1</b>
<b>1.2 ANTECEDENTES DEL PROBLEMA</b> .....	<b>3</b>
<b>1.3 PLANTEAMIENTO DEL PROBLEMA</b> .....	<b>5</b>
<b>1.4 PREGUNTAS DE INVESTIGACIÓN</b> .....	<b>6</b>
1.4.1 PREGUNTA GENERAL .....	6
1.4.2 PREGUNTAS ESPECÍFICAS.....	6
<b>1.5 OBJETIVOS DEL PROYECTO</b> .....	<b>6</b>
1.5.1 OBJETIVO GENERAL .....	6
1.5.2 OBJETIVOS ESPECÍFICOS .....	7
<b>1.6 JUSTIFICACIÓN</b> .....	<b>7</b>
<b>CAPÍTULO II - MARCO TEÓRICO</b> .....	<b>8</b>
<b>2.1 MACROENTORNO</b> .....	<b>8</b>
2.1.1 EVOLUCIÓN DE DEVOPS A NIVEL MUNDIAL .....	8
2.1.2 RELACIÓN DEVOPS - ÁGIL .....	11
2.1.3 TENDENCIAS TECNOLÓGICAS GLOBALES .....	15
2.1.4 NORMAS Y ESTÁNDARES INTERNACIONALES APLICABLES .....	16
2.1.4.1 AUTOMATIZACIÓN (CI/CD).....	16
2.1.4.2 CALIDAD.....	17
2.1.4.3 SEGURIDAD (DEVSECOPS).....	17
2.1.4.4 CUMPLIMIENTO NORMATIVO .....	18
2.1.4.5 GOBERNANZA DE TI.....	18
2.1.5 BUENAS PRÁCTICAS INTERNACIONALES.....	20
<b>2.2 MICROENTORNO - CONTEXTO NACIONAL (HONDURAS)</b> .....	<b>21</b>
2.2.1 PANORAMA DEL SECTOR TECNOLÓGICO EN HONDURAS .....	21
2.2.2 ESTADO DE ADOPCIÓN DE METODOLOGÍAS ÁGILES .....	21

2.2.3 OPORTUNIDADES DE MERCADO .....	21
2.2.4 OFERTA DE TALENTO TI EN EL PAÍS .....	23
2.2.5 LIMITACIONES Y RETOS NACIONALES .....	23
2.2.6 INFRAESTRUCTURA TECNOLÓGICA .....	24
2.2.7 CONTEXTO EMPRESARIAL - NUEVA GENERACIÓN DE SOFTWARE (NGS)24	
2.2.7.1 PERFIL DE LA EMPRESA .....	24
2.2.7.2 MISIÓN .....	24
2.2.7.3 VISIÓN .....	25
2.2.7.4 PRINCIPALES SERVICIOS.....	25
2.2.7.5 METODOLOGÍAS ACTUALES.....	26
2.2.7.6 ESTADO ACTUAL DE LA ENTREGA CONTINUA .....	26
2.2.7.7 OPORTUNIDADES ESPECÍFICAS .....	26
2.2.7.8 IMPACTO ESPERADO DE LA PROPUESTA .....	27
<b>2.3 TEORÍA DE SUSTENTO .....</b>	<b>27</b>
2.3.1 TEORÍA DE LA CUARTA REVOLUCIÓN INDUSTRIAL .....	28
2.3.2 TEORÍA DE LA INNOVACIÓN DISRUPTIVA (CHRISTENSEN).....	28
2.3.3 TEORÍA DE LA MEJORA CONTINUA (KAIZEN / CICLO PDCA) .....	29
<b>2.4 METODOLOGÍAS .....</b>	<b>30</b>
2.4.1 SCRUM COMO MARCO DE TRABAJO ÁGIL .....	30
2.4.1.1 FORTALEZAS DE SCRUM.....	32
2.4.1.2 LIMITACIONES DE SCRUM.....	33
2.4.1.3 ROLES.....	33
2.4.1.4 CICLO DE VIDA DE SCRUM.....	35
2.4.1.5 RETROSPECTIVA DEL SPRINT EN LA METODOLOGÍA SCRUM.....	36
2.4.2 DEVOPS COMO MODELO DE INTEGRACIÓN Y ENTREGA CONTINUA .....	36
2.4.2.1 PRINCIPIOS CULTURALES DE DEVOPS .....	37
2.4.3 ITIL V4 COMO MARCO DE GESTIÓN DE SERVICIOS TECNOLÓGICOS .....	38
<b>2.5 ANTECEDENTES DE LAS METODOLOGÍAS .....</b>	<b>40</b>
2.5.1 EXPERIENCIAS PREVIAS CON SCRUM .....	40
2.5.2 EXPERIENCIAS PREVIAS CON DEVOPS .....	41
2.5.3 EXPERIENCIAS PREVIAS CON ITIL 4 .....	42

<b>2.6 ANÁLISIS CRÍTICO DE LAS METODOLOGÍAS .....</b>	<b>45</b>
2.6.1 APLICABILIDAD DE SCRUM.....	45
2.6.1.1 REUNIONES EN EL SPRINT .....	46
2.6.2 APLICABILIDAD DE DEVOPS .....	47
2.6.3 APLICABILIDAD DE ITIL V4 .....	48
2.6.3.1 BENEFICIOS DE ITIL V4 EN ENTORNOS DEVOPS .....	48
2.6.3.2 DESAFÍOS Y LIMITACIONES DE ITIL V4 .....	49
2.6.3.3 PRÁCTICAS DE ITIL 4 Y SU APLICABILIDAD EN ENTORNOS DEVOPS.....	49
<b>2.7 INSTRUMENTOS UTILIZADOS .....</b>	<b>50</b>
2.7.1 HERRAMIENTAS PROPIAS DE SCRUM.....	50
2.7.2 HERRAMIENTAS Y PRÁCTICAS ASOCIADAS A DEVOPS .....	52
2.7.3 HERRAMIENTAS Y PRÁCTICAS DE ITIL V4.....	52
2.7.3.1 PRÁCTICAS DE ITIL COMO INSTRUMENTOS DE GESTIÓN .....	52
2.7.3.2 MATRIZ DE MADUREZ DE PRÁCTICAS ITIL V4 .....	53
<b>2.8 CONCEPTUALIZACIÓN .....</b>	<b>54</b>
<b>2.9 MARCO LEGAL Y NORMATIVO.....</b>	<b>57</b>
2.9.1 NORMATIVA INTERNACIONAL .....	57
2.9.1.1 REGULACIONES RELACIONADAS CON SCRUM .....	57
2.9.1.2 REGULACIONES APLICABLES A DEVOPS .....	58
2.9.1.3 REGULACIONES VINCULADAS A ITIL V4.....	58
2.9.2 NORMATIVA NACIONAL.....	58
2.9.2.1 MARCO NORMATIVO EN SCRUM .....	58
2.9.2.2 MARCO NORMATIVO EN ITIL V4.....	58
<b>CAPÍTULO III - METODOLOGÍA DE LA INVESTIGACIÓN.....</b>	<b>59</b>
<b>3.1 ENFOQUE .....</b>	<b>59</b>
<b>3.2 ALCANCE DE LA METODOLOGÍA .....</b>	<b>61</b>
<b>3.3 DISEÑO.....</b>	<b>61</b>
<b>3.4 POBLACIÓN.....</b>	<b>62</b>
<b>3.5 MUESTRA.....</b>	<b>63</b>
<b>3.6 TÉCNICA DE MUESTREO.....</b>	<b>65</b>
<b>3.7 CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN .....</b>	<b>65</b>

3.7.1 PERSONAL .....	65
3.7.2 PROCESOS.....	66
<b>3.8 HIPÓTESIS .....</b>	<b>67</b>
<b>3.9 OPERACIONALIZACIÓN DE LAS VARIABLES.....</b>	<b>68</b>
3.9.1 ESQUEMA DE VARIABLES DE ESTUDIO .....	69
3.9.2 MATRIZ DE OPERACIONALIZACIÓN DE LAS VARIABLES .....	70
<b>3.10 TÉCNICAS, INSTRUMENTOS, PROCEDIMIENTOS, ANÁLISIS DE LOS DATOS.....</b>	<b>72</b>
3.10.1 TÉCNICAS .....	72
3.10.2 INSTRUMENTOS .....	74
3.10.3 PROCEDIMIENTOS .....	76
3.10.4 ANÁLISIS DE LOS DATOS.....	77
<b>3.11 FUENTES DE INFORMACIÓN.....</b>	<b>77</b>
3.11.1 FUENTES PRIMARIAS.....	78
3.11.2 FUENTES SECUNDARIAS .....	79
<b>3.12 MATRIZ METODOLÓGICA.....</b>	<b>80</b>
<b>CAPÍTULO IV - RESULTADOS Y ANÁLISIS.....</b>	<b>83</b>
<b>4.1 INSTRUMENTOS Y PARTICIPANTES.....</b>	<b>83</b>
<b>4.2 DESCRIPCIÓN DE LOS RESULTADOS Y SU ANÁLISIS POR OBJETIVO .....</b>	<b>88</b>
4.2.1 ANÁLISIS DE LAS BRECHAS EN PROCESOS, HERRAMIENTAS Y COMPETENCIAS ACTUALES EN NGS .....	95
4.2.2 IDENTIFICACIÓN DE PRÁCTICAS Y HERRAMIENTAS DEVOPS APLICABLES EN NGS Y SU ALINEACIÓN CON ITIL V4.....	129
4.2.3 EVALUACIÓN DE LA ARQUITECTURA TECNOLÓGICA PARA LA IMPLEMENTACIÓN DEL FLUJO CI/CD.....	149
4.2.4 INDICADORES DE DESEMPEÑO Y SU IMPACTO EN LOS TIEMPOS ENTREGA CONTINUA DEL SERVICIO .....	171
<b>CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>199</b>
<b>5.1 CONCLUSIONES.....</b>	<b>199</b>
<b>5.2 RECOMENDACIONES.....</b>	<b>200</b>
<b>CAPÍTULO VI - APLICABILIDAD .....</b>	<b>202</b>

<b>6.1 MODELO DE INTEGRACIÓN DE PRÁCTICAS DEVOPS CON ENFOQUE ITIL V4 Y ALINEACIÓN ÁGIL MEDIANTE SCRUM PARA FORTALECER EL FLUJO CI/CD EN NGS.....</b>	<b>202</b>
<b>6.2 JUSTIFICACIÓN DE LA PROPUESTA.....</b>	<b>202</b>
<b>6.3 ALCANCE DE LA PROPUESTA.....</b>	<b>203</b>
6.3.1 OBJETIVOS .....	203
GENERAL.....	203
ESPECÍFICOS.....	203
<b>6.4 DESARROLLO DE LA PROPUESTA .....</b>	<b>203</b>
6.4.1 PLAN ESTRATÉGICO DE CIERRE DE BRECHAS PARA EL FLUJO CI/CD EN NGS.....	204
6.4.2 MODELO DE INTEGRACIÓN CI/CD INCORPORACIÓN DE PRÁCTICAS DEVOPS AL MARCO SCRUM Y ALINEACIÓN CON BUENAS PRÁCTICAS DE ITIL V4 .....	228
<b>6.5 MEDIDAS DE CONTROL .....</b>	<b>256</b>
6.5.1 PLAN DE INTEGRACIÓN DE MÉTRICAS DORA COMO PARTE DEL PIPELINE DE ENTREGA CONTINUA PARA UNA GESTIÓN BASADA EN DATOS.....	256
<b>6.6 CRONOGRAMA DE IMPLEMENTACIÓN .....</b>	<b>265</b>
<b>6.7 PRESUPUESTO E IMPACTO DEL PROYECTO.....</b>	<b>267</b>
<b>6.8 CONCORDANCIA DE LOS SEGMENTOS DE LA TESIS.....</b>	<b>271</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>274</b>
<b>ANEXOS.....</b>	<b>282</b>
<b>INSTRUMENTOS TEMÁTICOS.....</b>	<b>282</b>
ANEXO 1: ANÁLISIS FODA.....	282
ANEXO 2: MATRIZ DE MADUREZ DE PRÁCTICAS ITIL V4 EN NGS .....	282
<b>INSTRUMENTOS METODOLÓGICOS.....</b>	<b>284</b>
ANEXO 3: LA ENCUESTA.....	284
ANEXO 4: MATRIZ DE REVISIÓN DOCUMENTAL.....	284
ANEXO 5: MATRIZ COMPARATIVA .....	285
ANEXO 6: CHECKLIST TÉCNICO.....	286
<b>CARTA DE AUTORIZACIÓN DE LA EMPRESA O INSTITUCIÓN .....</b>	<b>287</b>

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Evolución histórica de DevOps desde la cultura Ágil (2008-2020).....	14
<b>Figura 2</b> Destaca el nearshoring en Honduras como uno de los mayores beneficiarios potenciales cuando se observan las perspectivas específicas de los países en la región (2024).....	22
<b>Figura 3</b> Teoría de la mejora Continua (Kaizen/ Ciclo PDCA).....	29
<b>Figura 4</b> Elementos principales en la creación de un producto.....	31
<b>Figura 5</b> Scrum taskboard.....	32
<b>Figura 6</b> Roles, artefactos y eventos principales.....	33
<b>Figura 7</b> Ciclo de carrera o de vida (Sprint) de Scrum.....	36
<b>Figura 8</b> Sistema de Valor de Servicio (Service Value System) de ITIL 4, con sus cinco componentes clave: principios rectores, gobernanza, cadena de valor del servicio, prácticas de gestión y mejora continua.....	39
<b>Figura 9</b> Cadena de Valor del Servicio (Service Value Chain) en el Sistema de Valor del Servicio (SVS) de ITIL 4.....	40
<b>Figura 10</b> Aplicabilidad de la metodología Scrum.....	46
<b>Figura 11</b> Aplicabilidad de DevOps: contexto, principios, ámbitos y limitaciones.....	47
<b>Figura 12</b> Gráfica Burndown.....	51
<b>Figura 13</b> ITIL Maturity Model.....	53
<b>Figura 14</b> Orden lógico para la metodología de investigación.....	59
<b>Figura 15</b> Elección del Diseño en un Enfoque Cualitativo.....	62
<b>Figura 16</b> Relación entre población y muestra en investigación.....	64
<b>Figura 17</b> Esquema de la variable independiente.....	69
<b>Figura 18</b> Esquema de la variable dependiente.....	70
<b>Figura 19</b> Clasificación de las fuentes de la información.....	78
<b>Figura 20</b> Mapa de relación entre variables, dimensiones y objetivos.....	82
<b>Figura 21</b> Relación de los objetivos específicos por cada instrumento aplicado.....	87
<b>Figura 22</b> Distribución por género del personal encuestado en NGS.....	88
<b>Figura 23</b> Distribución por rangos de edad de los participantes.....	89
<b>Figura 24</b> Nivel educativo alcanzado por los colaboradores participantes.....	90
<b>Figura 25</b> Años de experiencia en el área de TI o desarrollo del personal encuestado.....	91
<b>Figura 26</b> Cargo o rol desempeñado por los encuestados dentro de NGS.....	92
<b>Figura 27</b> Grado de familiaridad de los participantes con el término DevOps.....	93
<b>Figura 28</b> Grado de familiaridad de los participantes con el marco ITIL V4.....	94
<b>Figura 29</b> Análisis FODA de Procesos, herramientas y competencias actuales en NGS.....	96
<b>Figura 30</b> Documentación de procesos de desarrollo y despliegue.....	100
<b>Figura 31</b> Estandarización de los flujos de trabajo de desarrollo y entrega.....	101
<b>Figura 32</b> Claridad en roles y responsabilidades dentro del proceso CI/CD.....	102
<b>Figura 33</b> Adecuación de las herramientas utilizadas por el equipo.....	103
<b>Figura 34</b> Competencias del personal técnico para prácticas CI/CD y automatización.....	104
<b>Figura 35</b> Eficiencia del flujo actual de entrega continua.....	105
<b>Figura 36</b> Retrasos por actividades manuales en pruebas y despliegues.....	106
<b>Figura 37</b> Existencia de mecanismos automatizados de despliegue.....	107
<b>Figura 38</b> Configuración e integración de herramientas CI/CD.....	108
<b>Figura 39</b> Frecuencia de capacitación en nuevas herramientas o prácticas de automatización.....	109
<b>Figura 40</b> Impacto de la falta de documentación y estandarización en la calidad y velocidad de las	

entregas .....	110
<b>Figura 41</b> Aplicación de prácticas DevOps en desarrollo, pruebas y despliegue .....	112
<b>Figura 42</b> Nivel de adopción de pruebas automatizadas en el proceso de integración continua .....	114
<b>Figura 43</b> Colaboración entre equipos de desarrollo, QA y operaciones.....	115
<b>Figura 44</b> Disponibilidad de políticas formales para ramas, merge y revisión de código .....	116
<b>Figura 45</b> Uso de monitoreo y gestión de logs integrados en el pipeline con alertas .....	117
<b>Figura 46</b> Herramientas y plataformas utilizadas actualmente en los equipos para CI/CD .....	118
<b>Figura 47</b> Problemas identificados en el proceso de desarrollo en NGS .....	120
<b>Figura 48</b> Brechas principales en los procesos identificados en NGS .....	121
<b>Figura 49</b> Brechas tecnológicas identificadas en NGS .....	124
<b>Figura 50</b> Principales hallazgos sobre las herramientas en NGS.....	125
<b>Figura 51</b> Brechas principales de competencias identificadas.....	126
<b>Figura 52</b> Hallazgos principales sobre competencias .....	128
<b>Figura 53</b> Gestión formal de cambios .....	130
<b>Figura 54</b> Gestión de incidentes con SLA .....	131
<b>Figura 55</b> Trazabilidad entre cambios e incidentes.....	132
<b>Figura 56</b> Uso de flujos o plantillas aprobadas para despliegues .....	133
<b>Figura 57</b> Registros históricos de incidentes, cambios y versiones .....	134
<b>Figura 58</b> Prácticas ITIL consideradas mejor implementadas en NGS .....	135
<b>Figura 59</b> Prácticas DevOps identificadas para aplicar en NGS.....	139
<b>Figura 60</b> Sinergia entre prácticas, herramientas y políticas en el modelo DevOps.....	148
<b>Figura 61</b> Nivel de automatización en pruebas, despliegue y monitoreo .....	150
<b>Figura 62</b> Capacidad de escalabilidad de los entornos tecnológicos en NGS.....	151
<b>Figura 63</b> Integración de controles de seguridad en CI/CD.....	152
<b>Figura 64</b> Trazabilidad y registro de los despliegues.....	153
<b>Figura 65</b> Mecanismos de respaldo y recuperación ante fallos.....	154
<b>Figura 66</b> Actualización y documentación de la arquitectura tecnológica .....	155
<b>Figura 67</b> Disponibilidad de un repositorio central de arquitectura.....	156
<b>Figura 68</b> Uso de patrones y frameworks arquitectónicos en NGS .....	157
<b>Figura 69</b> Realización de revisiones técnicas y evaluaciones arquitectónicas.....	158
<b>Figura 70</b> Facilidad para integrar herramientas de monitoreo, CI/CD y control de versiones .....	159
<b>Figura 71</b> Disponibilidad y resiliencia de servicios críticos .....	160
<b>Figura 72</b> Uso de contenedores o servicios en la nube para escalar CI/CD.....	161
<b>Figura 73</b> Gestión segura de credenciales y accesos.....	162
<b>Figura 74</b> Nivel de madurez de la arquitectura en NGS .....	165
<b>Figura 75</b> Comparativa entre arquitectura monolítica vs. escalable con contenedores y microservicios	167
<b>Figura 76</b> Seguridad en el flujo actual CI/CD en NGS.....	168
<b>Figura 77</b> Trazabilidad en el flujo de actual de CI/CD en NGS .....	169
<b>Figura 78</b> Componentes clave de una arquitectura tecnológica moderna para flujos CI/CD bajo enfoque DevOps .....	170
<b>Figura 79</b> Principales hallazgos arquitectónicos del flujo CI/CD en NGS .....	171
<b>Figura 80</b> Mejora percibida en el tiempo promedio de entrega de versiones durante los últimos meses.	172
<b>Figura 81</b> Percepción de estabilidad en la calidad del software desplegado.....	173
<b>Figura 82</b> Nivel de conocimiento de los indicadores clave de desempeño (KPIs) por parte del equipo técnico.....	174
<b>Figura 83</b> Grado de satisfacción de los clientes con los entregables de software.....	175
<b>Figura 84</b> Frecuencia de despliegues o entregas de software en NGS.....	176
<b>Figura 85</b> Disponibilidad de tableros o reportes con indicadores de desempeño (KPIs y métricas DORA).	

.....	178
<b>Figura 86</b> Revisión periódica de los indicadores de desempeño por parte del equipo. ....	179
<b>Figura 87</b> Uso de los indicadores para identificar mejoras o apoyar decisiones de gestión. ....	180
<b>Figura 88</b> Medición actual de las métricas DORA en NGS. ....	181
<b>Figura 89</b> Tiempo de relación del cliente externo con NGS. ....	182
<b>Figura 90</b> Tipo de proyecto recibido por el cliente externo. ....	183
<b>Figura 91</b> Rol desempeñado por el cliente en el proyecto. ....	184
<b>Figura 92</b> Claridad en la planificación y cumplimiento de plazos de entrega. ....	185
<b>Figura 93</b> Calidad del software entregado (funcionalidad, estabilidad y requisitos). ....	186
<b>Figura 94</b> Velocidad y eficiencia en los despliegues (optimización de la entrega continua). ....	187
<b>Figura 95</b> Comunicación y seguimiento durante el ciclo de entrega. ....	188
<b>Figura 96</b> Capacidad de respuesta ante incidencias o cambios solicitados. ....	189
<b>Figura 97</b> Transparencia en los procesos y métricas compartidas con el cliente. ....	190
<b>Figura 98</b> Atención recibida por parte del equipo de NGS (profesionalismo y disposición). ....	191
<b>Figura 99</b> Nivel de satisfacción general con los servicios de NGS. ....	192
<b>Figura 100</b> Métricas DORA. ....	196
<b>Figura 101</b> Mapa de Calor de Riesgos - CI/CD. ....	213
<b>Figura 102</b> Plan de Acción para optimizar CICD. ....	215
<b>Figura 103</b> Políticas DevOps para estandarización CI/CD en el ciclo CI/CD. ....	234
<b>Figura 104</b> Integración de los eventos scrum y el ciclo DevOps. ....	241
<b>Figura 105</b> Modelo de las cuatro dimensiones de la gestión del servicio de ITIL 4. ....	244
<b>Figura 106</b> El Sistema de valor del servicio en ITIL 4. ....	247
<b>Figura 107</b> Resumen del plan de integración de Scrum - DevOps - ITIL V4. ....	254
<b>Figura 108</b> Diagrama de pipeline con integración de métricas DORA entre las diferentes fases. ....	258
<b>Figura 109</b> Uso Organizacional de las métricas DORA. ....	261
<b>Figura 110</b> Ejemplo de un tablero de métricas DORA. ....	262
<b>Figura 111</b> Revisión de indicadores. ....	263
<b>Figura 112</b> Diagrama de GANTT. ....	267

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Cuadro resumen de la evolución del DevOps a nivel mundial .....	10
<b>Tabla 2</b> Comparación de enfoques Ágil vs. DevOps .....	12
<b>Tabla 3</b> Problemas comunes en proyectos ágiles que DevOps aborda .....	13
<b>Tabla 4</b> Ámbitos clave de DevOps y ejemplos de estándares internacionales aplicables en cada uno.....	19
<b>Tabla 5</b> Síntesis de buenas prácticas internacionales en DevOps .....	20
<b>Tabla 6</b> Roles de la metodología Scrum .....	34
<b>Tabla 7</b> Fases del ciclo de vida de Scrum .....	35
<b>Tabla 8</b> Procesos y Prácticas de DevOps .....	38
<b>Tabla 9</b> Línea de tiempo de la evolución de ITIL .....	43
<b>Tabla 10</b> Prácticas de ITIL 4 y su aplicabilidad en entornos DevOps .....	50
<b>Tabla 11</b> Clasificación de las prácticas de ITIL v4.....	52
<b>Tabla 12</b> Definición del enfoque por cada objetivo específico .....	60
<b>Tabla 13</b> Distribución de la población y muestra por área.....	64
<b>Tabla 14</b> Resumen de la población y muestra de la investigación.....	65
<b>Tabla 15</b> Criterios de inclusión y exclusión según la muestra para el personal.....	66
<b>Tabla 16</b> Criterios de inclusión y exclusión según la muestra para procesos .....	66
<b>Tabla 17</b> Matriz de operacionalización de las variables .....	70
<b>Tabla 18</b> Relación entre técnicas, instrumentos, procedimientos y objetivos específicos .....	76
<b>Tabla 19</b> Plan de análisis cualitativo de los datos .....	77
<b>Tabla 20</b> Matriz de congruencia.....	80
<b>Tabla 21</b> Checklist de las herramientas, políticas y prácticas en NGS .....	98
<b>Tabla 22</b> Descripción y análisis de las respuestas abiertas .....	111
<b>Tabla 23</b> Descripción y análisis de las respuestas abiertas .....	119
<b>Tabla 24</b> Hallazgos principales en los procesos.....	122
<b>Tabla 25</b> Inventario visual de herramientas actuales en NGS.....	123
<b>Tabla 26</b> Competencias actuales y brechas identificadas.....	127
<b>Tabla 27</b> Resumen consolidado de hallazgos en procesos, herramientas y competencias.....	128
<b>Tabla 28</b> Descripción y análisis de las respuestas abiertas .....	136
<b>Tabla 29</b> Matriz Consolidada de Madurez de Prácticas ITIL V4 en NGS.....	137
<b>Tabla 30</b> Niveles ITIL.....	138
<b>Tabla 31</b> Compatibilidad y complementariedad entre Scrum y DevOps.....	140
<b>Tabla 32</b> Matriz de mapeo entre DevOps, Scrum e ITIL V4.....	142
<b>Tabla 33</b> Hallazgos principales sobre las prácticas identificadas para NGS.....	142
<b>Tabla 34</b> Herramientas DevOps identificadas.....	143
<b>Tabla 35</b> Evaluación Comparativa de Herramientas DevOps para Implementación CI/CD .....	145
<b>Tabla 36</b> Análisis de impacto y compatibilidad de políticas DevOps por fase del ciclo .....	146
<b>Tabla 37</b> Resumen de los hallazgos sobre políticas DevOps identificadas para NGS.....	147
<b>Tabla 38</b> Descripción y análisis de las respuestas abiertas. ....	162
<b>Tabla 39</b> Resumen de los resultados de las preguntas de la encuesta .....	164
<b>Tabla 40</b> Indicadores considerados más útiles para medir la eficiencia del proceso de entrega continua según la experiencia del personal encuestado.....	176
<b>Tabla 41</b> Aspectos que deben mejorarse según la experiencia del cliente en la entrega continua de software.....	193
<b>Tabla 42</b> Hallazgos sobre la ausencia de KPIs y su impacto en la entrega de valor CI/CD en NGS.....	194
<b>Tabla 43</b> Métricas DORA y su relación con las dimensiones del Objetivo 4.....	196

<b>Tabla 44</b>	Hallazgos sobre la gestión de KPIs y el potencial de adopción de métricas DORA en NGS...	197
<b>Tabla 45</b>	Resumen de hallazgos por objetivo específico .....	198
<b>Tabla 46</b>	Análisis y estrategias de mitigación de la Matriz FODA.....	206
<b>Tabla 47</b>	Glosario de siglas y términos usados en el plan.....	207
<b>Tabla 48</b>	Matriz de Mitigación Técnica para Herramientas, Políticas y Prácticas en NGS .....	208
<b>Tabla 49</b>	Matriz de criticidad para priorizar brechas con base al impacto y probabilidad.....	210
<b>Tabla 50</b>	Matriz del plan de acción (acciones, entregables y evidencia) .....	216
<b>Tabla 51</b>	Seguimiento y control del plan de acción .....	218
<b>Tabla 52</b>	Talleres para grupo selecto de desarrolladores .....	219
<b>Tabla 53</b>	Expansión y transferencia liderada por el Grupo Núcleo .....	220
<b>Tabla 54</b>	Checklist propuesto de adopción CI/CD por proyecto .....	221
<b>Tabla 55</b>	Matriz de mitigación para subir madurez ITIL v4 apoyándose en DevOps .....	222
<b>Tabla 56</b>	Ficha técnica del puesto DevOps.....	225
<b>Tabla 57</b>	Areas o roles del ingeniero DevOps .....	226
<b>Tabla 58</b>	Matriz de implementación de prácticas DevOps identificadas para NGS .....	229
<b>Tabla 59</b>	Plan de integración por etapas .....	231
<b>Tabla 60</b>	Tabla operativa de políticas por fase del ciclo DevOps .....	234
<b>Tabla 61</b>	Matriz RACI del modelo de políticas .....	235
<b>Tabla 62</b>	Checklist de verificación de adopción .....	237
<b>Tabla 63</b>	Políticas de control de versiones y beneficios para NGS.....	239
<b>Tabla 64</b>	Alineación del modelo Scrum con las prácticas DevOps en el ciclo del Sprint .....	242
<b>Tabla 65</b>	Alineación de roles Scrum con las prácticas DevOps del modelo propuesto .....	243
<b>Tabla 66</b>	Principios guía más 4 dimensiones aplicada a la integración DevOps en NGS.....	245
<b>Tabla 67</b>	Flujo de Valor: Entrega y despliegue controlado de software mediante CI/CD (NGS) .....	247
<b>Tabla 68</b>	Plan de integración DevOps ITIL v4 .....	250
<b>Tabla 69</b>	Matriz de lineamientos arquitectónicos para soportar CI/CD en NGS .....	252
<b>Tabla 70</b>	Matriz de las fases de la gestión del cambio organizacional.....	255
<b>Tabla 71</b>	Implementación por fases de las métricas DORA .....	257
<b>Tabla 72</b>	Métricas DORA integradas para NGS .....	259
<b>Tabla 73</b>	Integración técnica de métricas DORA por etapa del pipeline .....	260
<b>Tabla 74</b>	Cuadro de Capacitación y Adopción Organizacional para Métricas DORA en NGS .....	263
<b>Tabla 75</b>	Cronograma de implementación de actividades .....	265
<b>Tabla 76</b>	Costos de herramientas DevOps (USD).....	268
<b>Tabla 77</b>	Costos de consultoría externa DevOps .....	269
<b>Tabla 78</b>	Costos de infraestructura para CI/CD .....	269
<b>Tabla 79</b>	Resumen del presupuesto total estimado .....	270
<b>Tabla 80</b>	Impactos esperados del proyecto y contribución al ROI .....	270
<b>Tabla 81</b>	Concordancia de los segmentos de la tesis con la propuesta .....	271

# **CAPÍTULO I - PLANTEAMIENTO DE LA INVESTIGACIÓN**

## **1.1 INTRODUCCIÓN**

La presente investigación se refiere a la estandarización e implementación de prácticas DevOps integrándolas con metodologías ágiles y alineadas con las mejores prácticas de ITIL V4 en la consultora Nueva Generación de Software (NGS), para fortalecer la gestión de servicios de TI. Este método combina desarrollo y operaciones con la finalidad de mejorar la entrega de software, incrementar la calidad de los productos y servicios, y robustecer la capacidad de reacción frente a modificaciones o incidentes, todo dentro de un marco que garantice sostenibilidad y mejora continua.

El uso de prácticas ágiles y DevOps en NGS se realiza actualmente de manera parcial y no estándar, lo que da como resultado resultados inconsistentes y limita el aprovechamiento de los beneficios que estos enfoques ofrecen en conjunto. Esta investigación tiene como objetivo tratar esta problemática mediante el diseño de un marco formal que facilite la unificación de procesos, roles y métricas, asegurando de esta manera una eficiencia operativa y una cultura organizacional orientada a la calidad y a la mejora continua.

En el capítulo I se describe la dificultad que enfrenta la Consultora Nueva Generación de Software (NGS) en la gestión asociada a procesos manuales y a la ausencia de metodologías integradas que limiten las estimaciones de tiempo y la calidad del software. Por ello, se plantea una propuesta que combina metodologías ágiles, prácticas DevOps y lineamientos de ITIL V4 como solución para optimizar procesos, fortalecer la gestión de servicios y mejorar la calidad de los productos entregados a los clientes.

En el marco teórico, los cimientos conceptuales y contextuales de la investigación se desarrollan en el capítulo II. Se examinan las tendencias globales de la tecnología, las normas internacionales relacionadas con la calidad, la seguridad, el cumplimiento y la gobernanza de TI, así como las buenas prácticas a nivel internacional. También se estudia cómo DevOps ha evolucionado en todo el mundo y su vínculo con las metodologías ágiles e ITIL V4. Se evalúa el contexto del sector tecnológico en Honduras, el nivel de implementación de metodologías ágiles, la adopción de marcos de gestión de servicios, las oportunidades de mercado, la disponibilidad de talento, los desafíos y restricciones, además de la infraestructura tecnológica. Asimismo, se expone el entorno de la compañía Nueva Generación de Software (NGS), incluyendo su misión, visión,

servicios, métodos actuales, situación de la entrega continua y el efecto que se espera con la propuesta. Esta sección incluye el diseño de la metodología, la población y las muestras elegidas utilizando una muestra intencionada no probabilística, los criterios de inclusión y exclusión, y la operacionalización de las variables.

En el capítulo III se presenta la metodología de la investigación, en la cual se fundamenta la elección de un enfoque mixto con predominio cualitativo. Este enfoque se considera adecuado para analizar las brechas, oportunidades y posibilidades de integración de DevOps, metodologías ágiles e ITIL V4 en NGS. Este apartado presenta el diseño metodológico, así como la población y la muestra seleccionada a través de un muestreo no probabilístico intencionado. Se detallan los criterios de inclusión y exclusión, así como la operacionalización de las variables.

En este capítulo también se detallan las técnicas, instrumentos y procedimientos utilizados para la recolección de información, así como el plan de análisis de datos que combina la codificación, categorización, matrices comparativas y análisis FODA, todo orientado a construir la propuesta metodológica y definir los indicadores clave de desempeño (KPIs).

El Capítulo IV se exponen los resultados y su discusión. Se detallan los instrumentos utilizados y los participantes, y se estructura la discusión en función de cada objetivo específico. Se identifican las brechas en procesos, herramientas y habilidades del flujo CI/CD en NGS, la alineación entre DevOps e ITIL v4, la madurez de la arquitectura tecnológica para soportar automatización y entrega continua, y la situación de los indicadores de rendimiento asociados a la entrega de software.

En el Capítulo V se presentan las conclusiones y recomendaciones. Las conclusiones emanan de la comparación del marco teórico con los resultados, juzgando en qué medida se logran los objetivos específicos y se da respuesta a las preguntas de la investigación. Las recomendaciones se enfocan en las decisiones NGS para adoptar prácticas DevOps, fortalecer la arquitectura CI/CD, gestionar servicios bajo ITIL v4 y establecer una cultura de mejora continua basada en métricas.

Finalmente, en el Capítulo VI se aborda la aplicabilidad de la propuesta. Proponer un modelo de integración de prácticas DevOps con ITIL v4 y alineación ágil con Scrum para fortalecer el flujo CI/CD en NGS. En este capítulo se aborda el plan estratégico para cerrar brechas en procesos, herramientas y habilidades, el modelo de integración CI/CD con políticas operativas por etapa del ciclo DevOps, el plan para integrar métricas DORA al pipeline de entrega continua

y las directrices para su gestión basada en datos. de DORA en el flujo de trabajo de entrega continua y las directrices para su gestión basada en datos . También se incluye, un cronograma de implementación, la estimación de recursos necesarios y un análisis de costos e impacto que demuestren la viabilidad del modelo propuesto y su contribución al fortalecimiento de la gestión de servicios TI en la organización.

## **1.2 ANTECEDENTES DEL PROBLEMA**

En la última década, DevOps ha pasado de ser una práctica de nicho a un pilar esencial de la transformación digital, integrando desarrollo y operaciones para acelerar la entrega de software con calidad. Según el State of DevOps Report de Puppet, cerca del 80 % de las organizaciones ya aplican DevOps en sus procesos (Puppet, 2021). De forma similar, un estudio de Harvard Business Review Analytics halló que un 77 % de las empresas depende o planean depender de DevOps en el corto plazo. Gartner proyecta que para 2027 el 80 % de las compañías incorporarán plataformas DevOps integradas, frente a solo un 25 % en 2023 (Gartner, 2024).

Los beneficios de DevOps están respaldados por evidencia sólida; como el informe Accelerate State of DevOps de Google/DORA muestra que los equipos élite pasaron de representar un 7 % en 2018 a 26 % en 2021, logrando despliegues a demanda, tiempos de recuperación menores a una hora y tasas de fallo de cambios inferiores al 15 % (Forsgren et al., 2018). Estos equipos también reportan mayor rentabilidad, participación de mercado y satisfacción del cliente. Aproximadamente la mitad de las organizaciones que adoptan DevOps ya alcanzan altos niveles de desempeño, destacando mejoras en calidad de software y velocidad de entrega (Nagarro, 2022).

En Latinoamérica y Centroamérica, la adopción de DevOps es más reciente, aunque con un crecimiento sostenido. El Agile Adoption Report (2022) indica que un 15 % de las empresas en la región lo han implementado a nivel organizacional, un 50 % de forma parcial y un 28 % aún no cuentan con iniciativas DevOps. La región muestra avances significativos, ya que más de la mitad de los profesionales ya entienden DevOps como una cultura integral de personas, procesos y tecnología, no solo como un conjunto de herramientas.

Entre los factores que limitan su adopción destacan la baja conectividad fuera de zonas urbanas, la escasa inversión en tecnología y capacitación, la resistencia al cambio y la falta de talento especializado (AlfaPeople, 2024). Sin embargo, sectores como banca, gobierno, telecomunicaciones y servicios tecnológicos han liderado su implementación en la región

(DevSecOps Latam, 2020).

En Honduras, eventos como “Unlocking Cloud Potential” (2025), con participación de AWS y Tigo Business, han reforzado el papel de DevOps para acortar ciclos de innovación, mejorar el tiempo de salida al mercado y alinear tecnología con objetivos empresariales (La Prensa, 2025). Estas iniciativas demuestran un interés creciente por integrar DevOps como habilitador clave de la transformación digital.

En este panorama internacional y regional de creciente adopción de DevOps, la realidad de la consultora Nueva Generación de Software (NGS) se ubica en un punto de transición. La empresa ha incorporado en ciertos proyectos metodologías ágiles, principalmente Scrum, siguiendo en muchos casos los marcos de trabajo definidos por sus clientes. Sin embargo, en otros proyectos se continúa utilizando un enfoque propio que, si bien incluye elementos ágiles, no está plenamente estandarizado ni respaldado por documentación formal. También se han hecho intentos para usar herramientas como Docker y Kubernetes para manejar contenedores y organizar despliegues. Sin embargo, no hay guías, políticas ni procedimientos que regulen su uso, lo que limita su aprovechamiento y crea dependencia del conocimiento implícito.

Esta situación pone de manifiesto brechas de conocimiento y de gestión que impactan de forma directa en la eficiencia operativa y la calidad de los entregables. Aunque NGS posee talento técnico capaz, la ausencia de un marco formal para la adopción integral de prácticas DevOps provoca que una considerable parte de los procesos siga dependiendo de intervenciones manuales. No hay un flujo de integración continua (CI) ni de despliegue continuo (CD) completamente implementado, lo que aumenta el riesgo de errores en producción, retrabajos y lentitud en las respuestas ante problemas. Esta carencia también dificulta medir de manera objetiva el valor que podría aportar una transformación de este tipo, privando a la dirección de métricas para justificar inversiones o evaluar el impacto de los cambios.

Es crucial abordar estas brechas para que NGS se alinee con las mejores prácticas y responda a las demandas del mercado actual, que se centra cada vez más en la rapidez, calidad y adaptabilidad de las soluciones tecnológicas. Mantener los procesos actuales prolonga el riesgo de retrasos, incrementa los costos derivados de fallos y retrabajos, y puede erosionar la confianza de los clientes. En cambio, usar metodologías ágiles junto con prácticas DevOps traería beneficios claros, como ver en tiempo real el progreso de los proyectos, encontrar errores pronto, tomar

decisiones basadas en datos confiables, optimizar recursos, reducir riesgos y mejorar la satisfacción del cliente.

En un área donde la innovación continua es clave, la transformación de NGS hacia un modelo DevOps nítido y estandarizado podría hacer que sea un ejemplo en la región en el desarrollo de software de alta calidad.

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

Actualmente, las organizaciones tecnológicas se ven desafiadas al enfrentarse a mercados que ofrecen y solicitan la agilidad, calidad y capacidad de poderse adaptar constantemente. Es por ello por lo que al implementar metodologías ágiles y las prácticas, DevOps ha demostrado ser una vía para alcanzar estos retos, donde podemos reducir tiempos en las entregas, minimizando errores y, al final mejorando la satisfacción del cliente.

Estas tendencias y retos de la actualidad que se presentan a nivel internacional también se ven reflejadas en la realidad de la Consultora Nueva Generación de Software (NGS). Esta ha incorporado de forma parcial el marco de trabajo Scrum en algunos de los equipos, lo que se ve con la necesidad de implementar las metodologías ágiles. En este momento, la Consultora tiene un nivel de madurez bajo porque no tiene un modelo estandarizado ni una documentación formal. Esto hace que no se aplique de manera uniforme, lo que impide que los beneficios de estas metodologías tengan un impacto positivo y constante en todos los proyectos.

La falta de adoptar un marco único y transversal se deriva en procesos muy fragmentados y manuales, particularmente en la duración y entrega del software. Esta problemática la traducimos en los retrabajos frecuentes, errores en producción y una duración larga frente a la corrección de incidencias. A manera de estimación, podríamos decir que estas incidencias pueden representar del 15 % y un 25 % en el costo de los desarrollos, tomando en cuenta que las horas que se invierten en las correcciones a las incidencias, interrupción en los servicios y la pérdida de ser un equipo más productivo, ya que también los atrasos en los desarrollos o no llevar controlados estos tiempos pueden afectar la confianza en los clientes, que a menudo buscan soluciones rápidas. Más que un cambio a nivel tecnológico, NGS requiere un cambio cultural y organizacional. El introducir las prácticas DevOps permitiría ampliar los principios de agilísimo en toda la cadena de valor de la empresa.

En este contexto, surge la necesidad de analizar ¿cómo la implementación estructurada de

prácticas DevOps, integradas con metodologías ágiles y alineadas con el marco ITIL v4, puede optimizar la entrega continua de software en NGS, mejorando la eficiencia operativa, la calidad de los entregables y la satisfacción del cliente?

## **1.4 PREGUNTAS DE INVESTIGACIÓN**

### **1.4.1 PREGUNTA GENERAL**

¿Cómo la implementación estructurada de prácticas DevOps, integradas con metodologías ágiles y alineadas con las mejores prácticas de ITIL V4, puede optimizar la entrega continua de software en la Consultora Nueva Generación de Software (NGS), mejorando la eficiencia operativa, la calidad de los entregables y la satisfacción del cliente?

### **1.4.2 PREGUNTAS ESPECÍFICAS**

1. ¿Qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps?
2. ¿Qué prácticas, herramientas y políticas DevOps puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI?
3. ¿Qué arquitectura tecnológica es más adecuada para soportar la implementación de CI/CD en NGS, garantizando escalabilidad, seguridad y trazabilidad?
4. ¿Qué indicadores clave de desempeño (KPIs) deben definirse para medir el impacto de adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software, productividad y la satisfacción del cliente?

## **1.5 OBJETIVOS DEL PROYECTO**

### **1.5.1 OBJETIVO GENERAL**

Proponer la implementación estructurada de prácticas DevOps, integradas con metodologías ágiles y alineadas con las mejores prácticas de ITIL V4, para optimizar la entrega continua de software en la Consultora Nueva Generación de Software (NGS), mejorar la eficiencia operativa, la calidad de los entregables y la satisfacción del cliente.

### 1.5.2 OBJETIVOS ESPECÍFICOS

1. Analizar qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps.
2. Identificar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI.
3. Evaluar la arquitectura tecnológica más adecuada para soportar la implementación de un flujo de CI/CD en NGS, garantizando la escalabilidad, la seguridad y la trazabilidad.
4. Proponer un conjunto de indicadores clave de desempeño (KPIs) que permitan evaluar de manera objetiva el impacto de la adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software y satisfacción del cliente.

## 1.6 JUSTIFICACIÓN

Para la Consultora NGS, la propuesta de la estandarización e implementación de prácticas DevOps integrándolas con metodologías ágiles y alineadas con las mejores prácticas de ITIL V4, permitirá mejorar el proceso de entrega de software y ayudará a realizar de una manera adecuada la entrega continúa concentrándose los equipos en las actividades de mayor valor, aumentando la velocidad de los entregables. En la actualidad, el no tener un método estandarizado dificulta el desempeño en la entrega de los proyectos y genera retrasos o reprocesos que pueden generar la pérdida de clientes.

Al obtener métricas claras para tomar decisiones de manera confiable, esta propuesta también va en busca de que NGS tenga bien posicionado su lugar como una consultora moderna y ágil, fortaleciendo siempre su capacidad de ser líder en los proyectos con altos estándares de calidad. Además, al adoptar las prácticas de ITIL V4, refuerza la gobernanza y gestión de servicios de TI, garantizando que la oferta sea tecnológica, organizacional y estratégica.

## CAPÍTULO II - MARCO TEÓRICO

### 2.1 MACROENTORNO

#### 2.1.1 EVOLUCIÓN DE DEVOPS A NIVEL MUNDIAL

El movimiento DevOps surgió a fines de la década de 2000 como respuesta a las disfunciones en la industria del software provocadas por la separación rígida entre desarrollo y operaciones. A pesar de la adopción generalizada de metodologías ágiles, las dos áreas seguían trabajando en silos, con objetivos y responsabilidades aisladas. Los desarrolladores y personal de operaciones solían culparse mutuamente por retrasos en las entregas y fallos en producción. Fue entonces cuando líderes de ambas comunidades comenzaron a dialogar sobre una mejor forma de trabajar, sentando las bases de DevOps. En este período (2007-2008), profesionales como Patrick Debois, Gene Kim y John Willis iniciaron debates en foros y reuniones locales que cristalizaron en la idea de unir los equipos de desarrollo (Dev) y operaciones (Ops) bajo una misma cultura colaborativa.

La formalización del movimiento se atribuye al consultor belga Patrick Debois, quien acuñó el término DevOps en 2009 al organizar la primera conferencia DevOpsDays en Gante, Bélgica. Este evento inaugural inspirado en la famosa presentación “10+ Deploys per Day” de John Allspaw y Paul Hammond (Flickr) en la conferencia Velocity 2009, reunió a decenas de profesionales y marcó el nacimiento oficial de DevOps como concepto. En 2010, la conferencia DevOpsDays ya se había extendido a Estados Unidos con un evento en Mountain View, California, señal de la rápida propagación internacional de la cultura DevOps. Para 2012, la adopción creciente de estas prácticas motivó la creación del primer informe anual “State of DevOps” (Estado de DevOps) por Alanna Brown de Puppet Labs, con el fin de relevar las tendencias y beneficios de DevOps en organizaciones de todo el mundo. Al año siguiente, en 2013, la publicación de la novela empresarial “The Phoenix Project” popularizó aún más la filosofía DevOps al ilustrar de forma narrativa cómo una empresa ficticia transforma sus procesos de TI usando principios DevOps. Grandes compañías globales (como Amazon, Netflix, Target, Adobe) comenzaron a adoptar DevOps durante esta época, validando el movimiento en entornos corporativos reales.

A medida que DevOps demostraba su valor, su adopción a nivel mundial se aceleró. En 2014, aproximadamente un 16 % de las organizaciones reportaban haber implementado prácticas DevOps, porcentaje que fue creciendo año tras año. Para 2017, firmas de análisis declaraban que

“DevOps ha alcanzado velocidad de escape”, estimando que alrededor del 50 % de las organizaciones ya practicaban DevOps de alguna forma. De hecho, la consultora Forrester denominó 2017 como “el año de DevOps”, reflejando que la metodología había pasado de ser una tendencia emergente a un estándar en la industria de TI. En 2018, la adopción seguía en auge y se estimaba que cerca del 27 % de las organizaciones tenían equipos DevOps formalmente establecidos, comparado con el 16 % de 2014. La comunidad global también creció exponencialmente; ese año se celebraron más de 30 eventos DevOpsDays en distintas ciudades en todo el mundo, difundiendo las buenas prácticas y casos de éxito en diversos continentes. Cabe destacar que el interés académico acompañó esta expansión, el número de publicaciones científicas sobre DevOps aumentó de apenas 30 artículos en 2015 a 78 artículos en 2018 en la biblioteca digital de ACM, señal del creciente esfuerzo investigador por comprender y sistematizar DevOps. En resumen, para finales de la década de 2010, DevOps había pasado de ser un movimiento incipiente a consolidarse como una “mejor práctica” ampliamente aceptada a nivel mundial para mejorar la entrega continua de software.

En los años 2020 en adelante, DevOps se ha establecido como parte integral del macroentorno tecnológico global. Estudios recientes señalan que aproximadamente 4 de cada 5 organizaciones en el mundo ya aplican DevOps o planean hacerlo en el corto plazo, lo que indica que sus prácticas se han vuelto casi omnipresentes en la industria del software. Esta adopción masiva ha venido acompañada de la evolución y especialización del propio concepto de DevOps. Por un lado, han surgido enfoques derivados como DevSecOps (que integra la seguridad en la cultura DevOps) y DataOps/ML-Ops (aplicando principios DevOps a datos e inteligencia artificial). Por otro lado, la creciente complejidad de los entornos cloud nativos ha impulsado tendencias como la plataformización o Platform Engineering, en las cuales se crean plataformas internas para facilitar a los equipos de desarrollo y operaciones la entrega rápida y segura de software. Asimismo, la incorporación de inteligencia artificial en operaciones (AIOps) promete automatizar aún más la supervisión y respuesta a incidencias, reduciendo tiempos de detección y recuperación.

**Tabla 1** Cuadro resumen de la evolución del DevOps a nivel mundial

Año	Evento Clave
2007-2008	<ul style="list-style-type: none"> <li>Comunidades de desarrollo y operaciones denuncian una grave disfunción en el modelo tradicional de entrega de software, sentando las bases conceptuales de DevOps.</li> </ul>
2009	<ul style="list-style-type: none"> <li>Patrick Debois organiza en Bélgica la primera conferencia DevOpsDays, donde se acuña el término “DevOps”. El evento surge tras la charla “10 Deploys per Day” de Allspaw y Hammond (Flickr) en la VelocityConf, inspirando la unión de Dev y Ops.</li> </ul>
2010	<ul style="list-style-type: none"> <li>DevOpsDays se expande a EE. UU. con un evento en Mountain View, California, acelerando la difusión de DevOps en la comunidad global de TI.</li> </ul>
2012	<ul style="list-style-type: none"> <li>Puppet Labs publica el primer informe State of DevOps, iniciando la medición anual de la adopción y el impacto de DevOps en organizaciones de todo el mundo.</li> </ul>
2013	<ul style="list-style-type: none"> <li>Se publica <i>The Phoenix Project</i>, novela técnica de Gene Kim et al. que populariza DevOps al contar la transformación DevOps de una empresa ficticia.</li> <li>Ese año, destacadas empresas como Amazon y Netflix ya abrazan principios DevOps, validando el movimiento a nivel empresarial.</li> </ul>
2014	<ul style="list-style-type: none"> <li>El informe State of DevOps 2014 (Forsgren, Humble, Kim, et al.) reporta que la adopción de DevOps está “acelerándose”.</li> <li>El 16 % de las organizaciones encuestadas indica haber implementado DevOps.</li> <li>Inicia la publicación anual regular de estos reportes con miles de participantes globales.</li> </ul>
2017	<ul style="list-style-type: none"> <li>Forrester Research declara “el Año de DevOps” dado el ímpetu del movimiento; datos de la industria muestran que alrededor del 50 % de las organizaciones ya aplicaban DevOps para este año.</li> <li>DevOps se vuelve mainstream en TI, destacándose su rol central en la transformación digital empresarial.</li> </ul>

2018	<ul style="list-style-type: none"> <li>• DevOps alcanza una madurez significativa a nivel internacional, el 27 % de organizaciones reportan tener equipos DevOps consolidados.</li> <li>• Se celebran más de 30 conferencias DevOpsDays en todo el mundo, reflejando una comunidad global activa.</li> <li>• La investigación académica sobre DevOps también se multiplica (78 papers en 2018 vs 30 en 2015), señalando amplio interés en comprender sus prácticas y desafíos.</li> </ul>
2020+	<ul style="list-style-type: none"> <li>• DevOps se consolida mundialmente distintas encuestas indican que más del 75 % de las empresas han adoptado DevOps o planean hacerlo pronto.</li> <li>• Emergen prácticas especializadas como DevSecOps y se enfatiza la ingeniería de plataformas internas para escalar DevOps en grandes organizaciones.</li> <li>• Analistas proyectan que hacia 2025 - 2026, alrededor del 80 % de las empresas habrán incorporado equipos de plataformas o Platform Ops para respaldar DevOps.</li> <li>• La tendencia general muestra que DevOps, junto con sus variantes, continúa evolucionando e impregnando la industria tecnológica global como un componente fundamental de la entrega de software.</li> </ul>

Nota: Elaboración propia.

### 2.1.2 RELACIÓN DEVOPS - ÁGIL

Ágil y DevOps se han consolidado a nivel internacional como enfoques fundamentales en el desarrollo de software de nueva generación. Ambas metodologías comparten el objetivo de acelerar la entrega de valor al cliente con alta calidad. Según Alejos et al. (2024), “Ágil se centra en el desarrollo de software incremental e iterativo, mientras que DevOps se centra en la automatización integral del desarrollo y la entrega de software” (p. 232).

En otras palabras, Ágil optimiza el proceso de desarrollo (planificación, construcción y ajuste continuo del software), mientras que DevOps extiende estos principios a todo el ciclo de vida, integrando desarrollo y operaciones. De hecho, diversos autores sostienen que existe una estrecha relación entre ambas filosofías; incluso se considera que DevOps es una evolución del movimiento Ágil enfocada en eliminar la separación tradicional entre desarrollo y operaciones. Almeida et al. (2022) destacan al respecto:

Ambas propuestas están relacionadas. DevOps puede entenderse como una evolución del desarrollo ágil, que se concibió gradualmente a través de experiencias prácticas de implementación en el desarrollo de software. Considerando que el método ágil tiene su enfoque dirigido específicamente al desarrollo de software, y que DevOps pretende involucrar al área de desarrollo de software en la implementación y operación del software desarrollado, entonces, los procesos DevOps se pueden implementar dentro de los procesos Ágiles (Almeida et al., 2022).

Esta relación DevOps-Ágil se manifiesta en que DevOps complementa y potencia las prácticas ágiles existentes. Ágil establece un modelo de entregas rápidas y frecuentes alineadas con las necesidades del cliente, mientras que DevOps lo optimiza integrando infraestructura, automatización y operaciones en ese flujo continuo. En la práctica, los principios ágiles (colaboración, retroalimentación constante, equipos autoorganizados, etc.) están firmemente arraigados en la cultura DevOps, la cual promueve una comunicación y colaboración aún más estrechas entre todos los equipos de TI. Gracias a esto, DevOps permite hacer despliegues más frecuentes y confiables. Esto se logra mediante prácticas como integración continua (CI), entrega continua (CD), pruebas automatizadas e infraestructura como código, que garantizan calidad y rapidez sin perder estabilidad. De hecho, autores como Leite et al. (2019) describen a DevOps como “un esfuerzo organizacional colaborativo y multidisciplinario para automatizar la entrega continua de nuevas actualizaciones de software garantizando su corrección y confiabilidad” (p.1), reflejando cómo amplía los valores ágiles hacia la fase de despliegue y operación.

**Tabla 2** Comparación de enfoques Ágil vs. DevOps

Aspecto	Metodologías Ágiles	Enfoque DevOps
Alcance de colaboración	Colaboración principalmente entre el equipo de desarrollo y el cliente durante el desarrollo.	Colaboración estrecha entre los equipos de desarrollo y operaciones (IT) en todas las etapas.
Objetivo Principal	Entregas incrementales rápidas de software funcional al cliente; alta adaptabilidad a cambios.	Entregas continuas de software en producción con alta automatización, manteniendo estabilidad y calidad.
Prácticas Claves	Iteraciones cortas ( <i>sprints</i> ), feedback frecuente del usuario, integración de cambios sobre la marcha (Agile Alliance, 2001, p.1).	Integración continua (CI), despliegue continuo (CD), infraestructura como código, monitoreo y pruebas automatizadas (Ebert <i>et al.</i> , 2016, p.95).

Cultura	Valores de colaboración con el cliente, equipos autoorganizados, respuesta rápida al cambio, mejora continua (Manifiesto Ágil, 2001, p.1).	Cultura de responsabilidad compartida entre dev y ops, eliminación de silos, énfasis en automatización y monitoreo para acelerar entregas sin sacrificar calidad.
---------	--	---

Nota: Elaboración propia con base en Domínguez Acosta, 2021, Ebert et al., 2016.

Es fundamental señalar que DevOps surgió en respuesta a ciertas limitaciones de las metodologías ágiles tradicionales en entornos reales de TI. Por ejemplo, durante la conferencia Agile 2008 se introdujo el concepto de “infraestructura ágil”, reconociendo la necesidad de mayor interacción entre desarrollo y operaciones, evento histórico que sembró la semilla de DevOps.

**Tabla 3** *Problemas comunes en proyectos ágiles que DevOps aborda*

Problema en Ágil	Solución con DevOps	Beneficio principal
Retrasos en lanzamientos: las nuevas funcionalidades a veces se demoran en llegar al usuario final.	Automatización de pruebas e implementación de despliegues continuos.	Reducción del time to market mediante la liberación inmediata de cada incremento.
Integración complicada: los componentes desarrollados en paralelo pueden presentar incompatibilidades.	Prácticas de integración continua y pruebas automatizadas desde etapas tempranas.	Mayor compatibilidad y estabilidad del software durante todo el ciclo de desarrollo.
Aseguramiento de calidad insuficiente: bajo presión de entregas rápidas, no siempre se garantiza la calidad.	Automatización de la calidad en la tubería DevOps (pruebas unitarias, de integración, seguridad, etc.).	Detección temprana de errores y entregas más confiables con alta calidad.

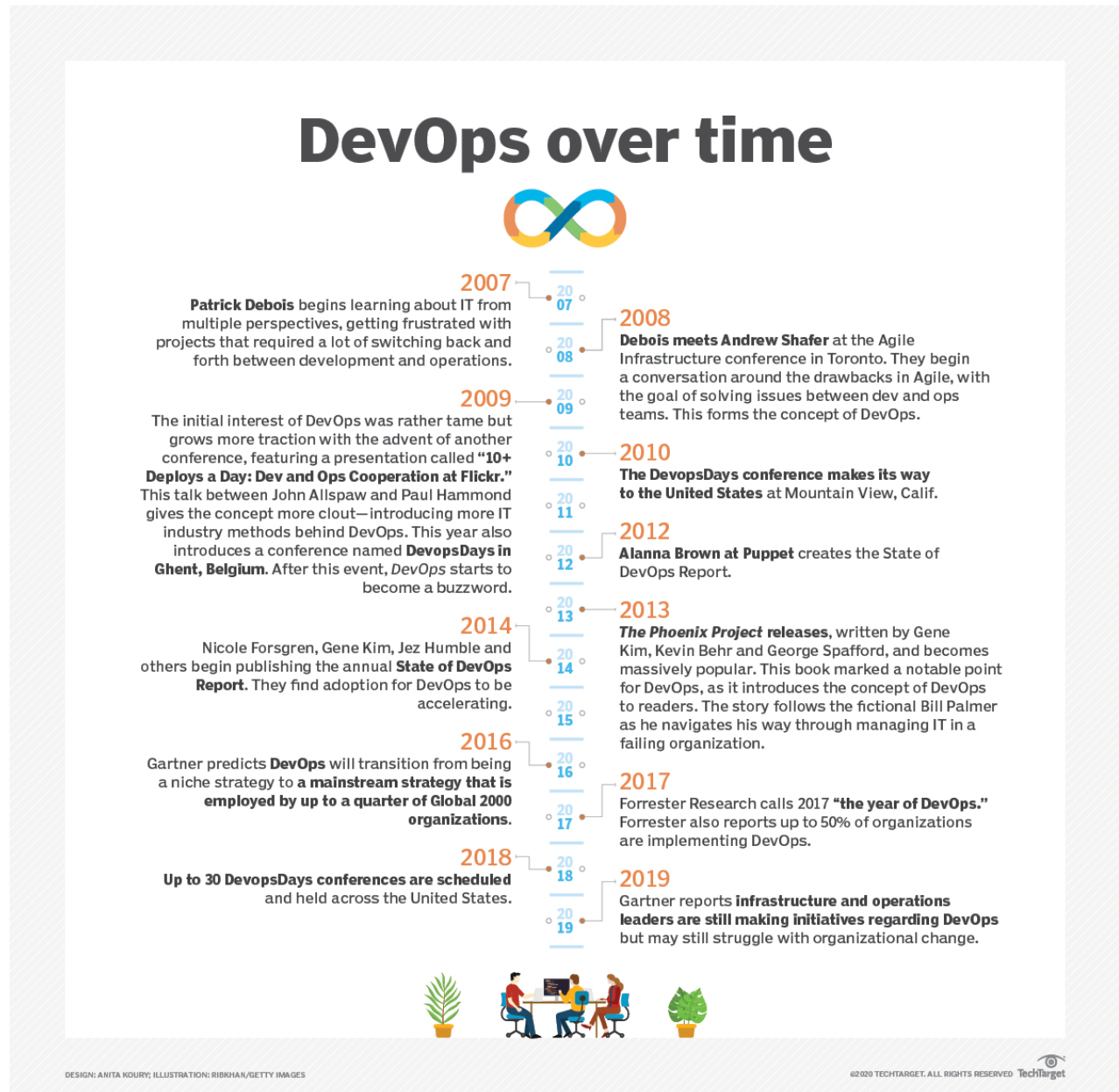
Nota: Adaptado de Desafíos en la adopción de un enfoque DevOps para el desarrollo y las operaciones de software (Hamunen & Joonas, 2016).

Estas prácticas conjuntas Agile-DevOps impulsan también una cultura de responsabilidad compartida. Por ejemplo, DevOps promueve que desarrolladores y personal de operaciones trabajen con objetivos unificados y se reparten responsabilidades en lugar de operar en silos separados. En última instancia, al incorporar DevOps sobre un fundamento ágil, las organizaciones logran equipos multifuncionales con mayor colaboración, visibilidad y capacidad de adaptación. Esto se traduce en beneficios tangibles. Estudios recientes han identificado más de una decena de beneficios al combinar Ágil + DevOps, entre los cuales destacan la reducción del ciclo de entrega (entregas más rápidas al cliente), el aumento de la automatización de procesos, la mejora en la

comunicación entre equipos y una mayor calidad del software liberado.

Por ejemplo, en empresas que han usado DevOps en lugar de metodologías ágiles, se nota que el tiempo para hacer cambios en producción se acorta mucho, y la cantidad de errores y trabajo extra disminuye debido a las pruebas continuas y el monitoreo integrado (Almeida et al., 2022, p.5).

**Figura 1** Evolución histórica de DevOps desde la cultura Ágil (2008-2020)



Fuente: Adaptado de The history of DevOps: A visual timeline, por Alexander S. Gillis, 2020.

### 2.1.3 TENDENCIAS TECNOLÓGICAS GLOBALES

El macroentorno tecnológico internacional se caracteriza por cambios acelerados e innovaciones disruptivas que están transformando todos los sectores productivos. Estudios recientes indican que la tecnología se ha convertido en el principal factor de cambio y disrupción para las empresas a nivel global. En particular, avances en inteligencia artificial (IA), computación en la nube, automatización robótica e Internet de las Cosas (IoT) están presentando un sinnúmero de oportunidades para mejorar procesos y modelos de negocio en todo el mundo (Foro Económico Mundial, 2024, p.2). Por ejemplo, la adopción masiva de IA generativa está revolucionando industrias como la salud, las finanzas y el marketing, permitiendo desde la creación de contenido hasta el análisis predictivo avanzado. Asimismo, la ciberseguridad se ha vuelto un foco central debido al creciente volumen y sofisticación de ciberataques globales, impulsando inversiones en tecnologías de protección de datos y resiliencia digital. Otras tendencias clave incluyen la proliferación de gemelos digitales (digital twins) para simulación en tiempo real, la convergencia de realidad virtual/aumentada con aplicaciones empresariales, y un énfasis transversal en la sostenibilidad tecnológica, es decir, el desarrollo de soluciones TIC energéticamente eficientes y orientadas a combatir el cambio climático.

En el ámbito del desarrollo de software y la gestión de TI, destaca la adopción global de metodologías ágiles y prácticas DevOps como tendencia tecnológica y organizacional. A nivel mundial, las organizaciones de software han incorporado DevOps como estrategia para mantenerse competitivas, reduciendo tiempos de entrega sin comprometer la calidad (García-Mireles et al., 2024, p.38). De hecho, más del 80 % de las empresas de tecnología reportan haber implementado alguna forma de DevOps en sus procesos (ISACA, 2022, p.5), aunque muchas aún se encuentran en etapas iniciales de madurez en esta práctica.

Paralelamente, las metodologías ágiles han pasado de ser una novedad para convertirse en el estándar dominante para la gestión de proyectos de software en la última década. Encuestas internacionales indican que un porcentaje alto de organizaciones (alrededor del 90 %) ya utilizan enfoques ágiles en alguno de sus equipos de desarrollo (Digital.ai, 2022, p.3). Esta tendencia hacia marcos ágiles/DevOps combinados responde a la presión global por lograr una “transformación digital” efectiva. Las empresas buscan ser más rápidas, eficientes y orientadas al cliente, lo cual

requiere tanto tecnologías innovadoras como procesos internos ágiles.

En suma, el contexto internacional en materia tecnológica está definido por una convergencia de tendencias globales, por un lado, la emergencia de nuevas tecnologías disruptivas (IA, automatización, realidad extendida, computación al borde, etc.) que abren oportunidades sin precedentes; y, por otro lado, la evolución de metodologías y prácticas en la industria del software (Agile, DevOps, DevSecOps, ingeniería de plataforma, entre otros) que permiten a las organizaciones aprovechar dichas tecnologías de manera ágil. Los líderes empresariales a nivel mundial reconocen que adoptar estas tendencias no es opcional, sino necesario quien logra integrar innovación tecnológica con agilidad operacional obtiene ventajas competitivas significativas, pudiendo responder más rápidamente a las cambiantes demandas del mercado. Por ello, se espera que en los siguientes años continúe acelerándose la inversión global en iniciativas de transformación digital, automatización inteligente y cultura DevOps/Ágil, configurando un macroentorno donde la capacidad de adaptación y la rapidez de entrega serán factores críticos de éxito en prácticamente todas las industrias (Contreras, 2024, p.2).

#### 2.1.4 NORMAS Y ESTÁNDARES INTERNACIONALES APLICABLES

En el contexto internacional, las organizaciones que adoptan DevOps se apoyan en normas y estándares reconocidos globalmente para asegurar la calidad, seguridad, cumplimiento y gobernanza de sus procesos. Hay marcos de mejores prácticas y estándares internacionales que orientan la automatización continua, la gestión de calidad, la seguridad de la información, el cumplimiento de normas y la gobernanza de TI en ambientes DevOps. A continuación, se destacan los principales referentes en cada ámbito:

##### 2.1.4.1 AUTOMATIZACIÓN (CI/CD)

Un hito importante es la publicación del estándar IEEE 2675-2021 / ISO/IEC/IEEE 32675:2022, el primer estándar internacional específico de DevOps. Esta norma “proporciona requisitos y guía sobre la implementación de DevOps para definir, controlar y mejorar los procesos del ciclo de vida del software, aplicándose dentro de una organización o proyecto para construir, empaquetar y desplegar software y sistemas de forma segura y confiable” (ISO, 2022, p. vii). Dicho estándar enfatiza principios técnicos como “continuous everything” (automatización continua) y el establecimiento de controles de TI efectivos (IEEE, 2021, p. 3).

En otras palabras, a nivel internacional se fomenta la integración continua y la entrega

continua como prácticas fundamentales. Por ejemplo, se sugieren herramientas de pipeline CI/CD e infraestructura como código para automatizar la entrega de software de manera consistente. Estas prácticas se alinean con estándares de procesos de software preexistentes (como ISO/IEC 12207) y reflejan el principio DevOps de “shift-left”, integrando controles de calidad y seguridad desde las primeras fases del desarrollo (Ramaj et al., 2022, p. 2).

#### 2.1.4.2 CALIDAD

En cuanto a gestión de la calidad, las organizaciones suelen adherirse a estándares de sistemas de calidad como ISO 9001:2015 (gestión de calidad) para asegurar la mejora continua de procesos. Si bien ISO 9001 es genérica, su enfoque en la satisfacción del cliente y mejora de procesos complementa a DevOps al fomentar ciclos de retroalimentación cortos y entregas fiables. Asimismo, el modelo Capability Maturity Model Integration (CMMI) se considera un referente internacional que guía la madurez de procesos de desarrollo y entrega de software (Ramaj et al., 2022, p. 2).

En entornos DevOps, mantener la calidad implica integrar pruebas automatizadas en el pipeline (por ejemplo, pruebas unitarias, integración, testing continuo) y monitorear métricas de rendimiento y calidad de código (p. ej., mediante SonarQube), alineándose con estándares de calidad de producto (como ISO/IEC 25010) y asegurando que cada liberación cumpla con los criterios de calidad acordados (Jenkins & Smith, 2024, p. 15).

#### 2.1.4.3 SEGURIDAD (DEVSECOPS)

La seguridad de la información es primordial en DevOps global. El estándar ISO/IEC 27001:2022 (Sistema de Gestión de Seguridad de la Información) es ampliamente adoptado para certificar que se siguen controles rigurosos de seguridad. De hecho, “ISO/IEC 27001 contiene un anexo dedicado al cumplimiento de requisitos legales y contractuales, con el objetivo de evitar brechas a obligaciones legales relacionadas con la seguridad de la información” (Ramaj et al., 2022, p. 5). Esto muestra que la seguridad y el cumplimiento están entrelazados, las prácticas DevOps deben asegurar la confidencialidad, integridad y disponibilidad de los sistemas conforme a estándares internacionales.

La integración de seguridad en el pipeline (DevSecOps) permite cumplir con normas como ISO 27001 de forma automatizada. Por ejemplo, incorporar análisis de vulnerabilidades, escaneos de código (SAST/DAST) y gestión de parches continuos en el flujo DevOps facilita “el

cumplimiento de normativas como GDPR o ISO 27001” (Decide, 2025, p. 4). Según un estudio reciente, la automatización en DevOps habilita la integración de controles de seguridad y cumplimiento sin ralentizar el proceso de entrega (Abrahams & Langerman, 2021, p. 59). Esto respalda la noción de Compliance as Code, donde las políticas de seguridad (por ejemplo, reglas OWASP, requisitos de cifrado) se implementan como código en la infraestructura, garantizando que cada despliegue sea seguro y conforme a estándares (Ramaj et al., 2022, p. 7).

#### 2.1.4.4 CUMPLIMIENTO NORMATIVO

En el ámbito del cumplimiento (compliance) existen tanto regulaciones globales como estándares de sistemas de cumplimiento. Un ejemplo es el Reglamento General de Protección de Datos (GDPR) de la UE, cuya observancia suele incorporarse en pipelines DevOps para tratamiento de datos personales (Ramaj et al., 2022, p. 3).

El incumplimiento de tales regulaciones conlleva sanciones severas. Por ello, se promueve una cultura DevOps consciente de cumplimiento continuo. La norma ISO 37301:2021 (Sistemas de gestión de compliance) refuerza esta visión al concebir el cumplimiento como “un proceso continuo enfocado en monitorear que la organización cumpla con sus obligaciones” (Ramaj et al., 2022, p. 4). En la práctica, muchas organizaciones integran controles de cumplimiento en las fases DevOps, por ejemplo, comprobaciones automáticas de configuraciones contra estándares (como PCI DSS en finanzas) o políticas de privacidad integradas en el código.

Un estudio sistemático destaca que “las actividades de auditoría y cumplimiento pueden automatizarse e integrarse en los pipelines DevOps para facilitar el proceso de compliance” (Ramaj et al., 2022, p. 4). Esto significa que herramientas CI/CD pueden ejecutar pruebas de cumplimiento de forma continua, reduciendo intervención humana y errores, a la vez que agilizando las auditorías (Ramaj et al., 2022, p. 4).

#### 2.1.4.5 GOBERNANZA DE TI

La gobernanza en entornos DevOps se apalanca en marcos internacionales que aseguran control y alineación con objetivos de negocio. Por ejemplo, ITIL 4 (2020) brinda prácticas para gestión de servicios de TI alineadas con DevOps, e ISO/IEC 20000-1:2018 es la norma internacional para gestión de servicios de TI (IT Service Management), aplicable a operaciones DevOps.

Estudios indican que en la industria se usan marcos de trabajo como ITIL y COBIT, junto

con estándares internacionales como ISO/IEC 29110 y la norma ISO/IEC 20000 (Orozco et al., 2022, p. 425), para crear procesos consistentes en la gestión de infraestructura y servicios. COBIT 2019 (ISACA) es otro marco ampliamente reconocido que provee principios de gobernanza de TI y control de procesos, complementando la agilidad de DevOps con controles corporativos claros. Asimismo, la norma ISO/IEC 38500:2015 de gobierno corporativo de TI ofrece directrices para que la alta dirección supervise eficazmente las iniciativas de TI (incluyendo DevOps) con accountability y cumplimiento.

En la práctica, implementar DevOps con gobernanza significa definir políticas claras (por ejemplo, criterios de segregación de ambientes, aprobaciones de cambios, gestión de riesgos) y medir la madurez DevOps con métricas alineadas a objetivos del negocio. De este modo se equilibra la velocidad que aporta DevOps con la controlabilidad que exige la gobernanza corporativa (Shahin et al., 2020, p. 8).

**Tabla 4** *Ámbitos clave de DevOps y ejemplos de estándares internacionales aplicables en cada uno*

Ámbito	Estándares/Marcos Internacionales Relevantes
<b>Automatización (CI/CD)</b>	<ul style="list-style-type: none"> <li>• IEEE 2675-2021 / ISO/IEC 32675:2022 - Estándar DevOps (automatización de build, entrega)</li> <li>• ISO/IEC 12207 - Procesos de ciclo de vida de software (integrado con DevOps)</li> </ul>
<b>Calidad</b>	<ul style="list-style-type: none"> <li>• ISO 9001:2015 - Gestión de calidad (mejora continua en procesos)</li> <li>• CMMI Dev - Modelo de madurez de capacidades (calidad y procesos de desarrollo)</li> </ul>
<b>Seguridad</b>	<ul style="list-style-type: none"> <li>• ISO/IEC 27001:2022 - Gestión de seguridad de la información (controles de seguridad)</li> <li>• OWASP, NIST CSF - Marcos internacionales de seguridad aplicables en DevSecOps</li> </ul>
<b>Cumplimiento</b>	<ul style="list-style-type: none"> <li>• ISO 37301:2021 - Sistemas de gestión de compliance (cumplimiento continuo)</li> <li>• GDPR (UE 2016), PCI DSS - Regulaciones y estándares sectoriales integrados en DevOps</li> </ul>

<b>Gobernanza de TI</b>	<ul style="list-style-type: none"> <li>• ISO/IEC 20000-1:2018 - Gestión de servicios TI (ITSM) para operaciones DevOps</li> <li>• COBIT 2019, ISO/IEC 38500:2015 - Gobernanza y control de TI alineados con DevOps</li> </ul>
-------------------------	---

Nota: Elaboración propia.

### 2.1.5 BUENAS PRÁCTICAS INTERNACIONALES

A escala global, la literatura académica converge en que las buenas prácticas de DevOps combinan automatización de extremo a extremo, integración/entrega/despliegues continuos (CI/CD), arquitecturas modularizadas, pruebas automatizadas, observabilidad/monitorización continua e infraestructura como código, todo ello sostenido por una cultura de colaboración que elimina silos entre desarrollo y operaciones. Una síntesis influyente es el mapa conceptual de Leite et al. (2019), que describe DevOps como “un esfuerzo colaborativo y multidisciplinario para automatizar la entrega continua de nuevas versiones garantizando su corrección y confiabilidad”, y destaca la relación práctica entre automatización, arquitectura (débilmente acoplada), contenedorización y monitorización en producción.

**Tabla 5** Síntesis de buenas prácticas internacionales en DevOps

<b>Práctica núcleo</b>	<b>Síntesis de la evidencia</b>
CI/CD automatizado y por etapas	Reduce tiempos de construcción/prueba y riesgo de despliegue; eleva la confiabilidad del proceso. (Shahin et al., 2017)
Arquitectura modular/ “loosely coupled”	Favorece despliegues frecuentes y seguros; facilita automatización. (Shahin et al., 2017; Leite et al., 2019)
Infraestructura como código + Contenedores	Estandariza entornos y reduce errores manuales en operaciones. (Leite et al., 2019)
Monitorización/observabilidad y “run-time” SRE	Mejora el desempeño, disponibilidad y resiliencia post-despliegue. (Leite et al., 2019)

Nota: Adaptado de "Continuous Integration, Delivery and Deployment".

## **2.2 MICROENTORNO - CONTEXTO NACIONAL (HONDURAS)**

### **2.2.1 PANORAMA DEL SECTOR TECNOLÓGICO EN HONDURAS**

Honduras avanza en su hoja de ruta digital con el Plan Nacional de Gobierno Digital 2023-2026, que establece prioridades en interoperabilidad, servicios en línea y formulación de una estrategia nacional de ciberseguridad (Gobierno de Honduras, 2023). No obstante, persisten brechas. Según la Unión Internacional de Telecomunicaciones, en 2023 solo el 58.3 % de la población utilizaba Internet y apenas el 51.4 % de los hogares contaba con conexión fija, lo que refleja disparidad entre áreas urbanas y rurales (ITU, 2023). Estas condiciones influyen en la adopción de prácticas avanzadas como DevOps, dado que la madurez tecnológica nacional es aún desigual.

### **2.2.2 ESTADO DE ADOPCIÓN DE METODOLOGÍAS ÁGILES**

Aunque la evidencia empírica y concreta acerca de la implementación de ágil en las compañías nacionales no es abundante, los planes de transformación digital y la preparación para obtener certificaciones están promoviendo un aumento progresivo. Esto queda evidenciado gracias a trabajos académicos recientes y a la oferta formativa local (por ejemplo, cursos de ITIL/Ágil en universidades y proveedores). En los programas de posgrado se están implementando marcos escalados (SAFe) y se ofrece educación continua en Scrum/Kanban. Esto muestra el interés de las empresas y las instituciones en las prácticas ágiles y su combinación con DevOps/ITSM.

A nivel regional, estudios muestran que el 87 % de las empresas latinoamericanas ya implementan metodologías ágiles en sus proyectos de TI, aunque en muchos casos de forma parcial o sin un marco estandarizado (Gartner, 2023). En Honduras ocurre lo mismo; prácticas como Scrum y Kanban han sido adoptadas por varias consultoras, pero en niveles de madurez heterogéneos. Informes globales como el de DORA/Google (2024) confirman que los mayores beneficios se alcanzan cuando Agile se conecta con DevOps, creando un flujo de integración y entrega continua, medible y sostenible.

### **2.2.3 OPORTUNIDADES DE MERCADO**

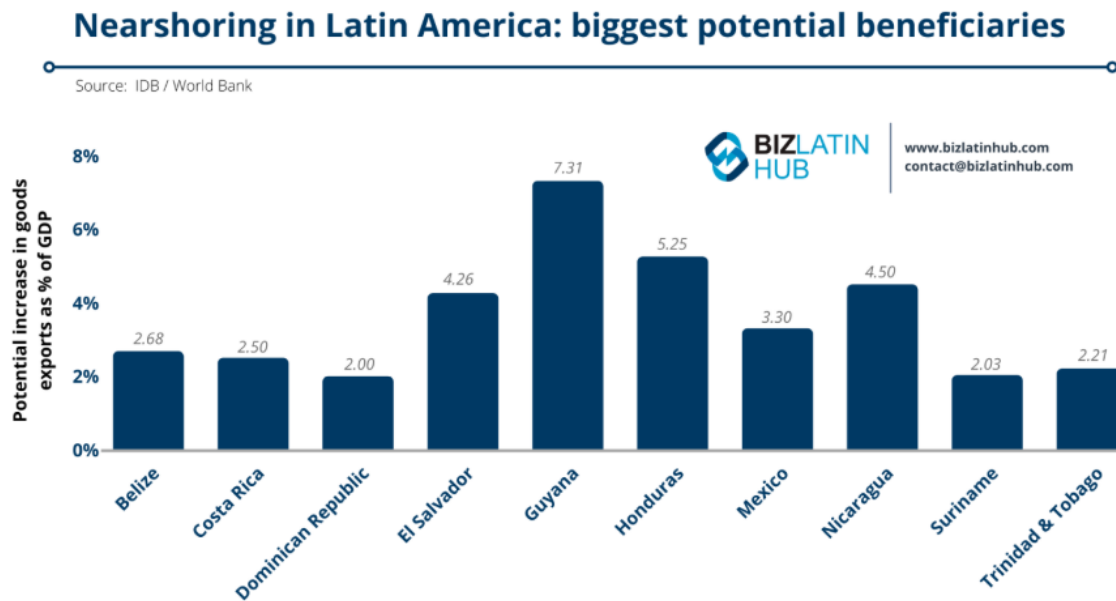
En toda la región Centroamérica, los países que conforman el triángulo Norte (El Salvador, Guatemala y Honduras) están siendo posicionados como un atractivo hub de nearshoring por empresas de los Estados Unidos. Siendo atraída la parte de la población joven que adopta servicios digitales, implementando opciones estratégicas para el diseño de las cadenas de suministros y

operatividad de TI. Mediante inversión por medio de Alianza para Centroamérica, se ha visto el interés por invertir en desarrollar el talento de la región.

Podemos decir que Honduras, desde la perspectiva nearshoring en TI, ofrece múltiples ventajas competitivas:

- Ubicación estratégica y zona horaria compatible con EE. UU.
- Ahorro significativo de costos
- Talento calificado, creciente bilingüismo y familiaridad cultural, lo que reduce la fricción en equipos distribuidos.
- Infraestructura y conectividad mejoradas

**Figura 2** Destaca el nearshoring en Honduras como uno de los mayores beneficiarios potenciales cuando se observan las perspectivas específicas de los países en la región (2024).



Fuente: Tomado de Biz Latin Hub (2023).

Parte de las industrias que han permitido el Nearshoring en Honduras es la Tecnología de la Información y Desarrollo de Software. Este sector ha experimentado un gran crecimiento, ya que se cuenta con una gran mano de obra calificada, dominando el lenguaje y las metodologías de desarrollo.

#### 2.2.4 OFERTA DE TALENTO TI EN EL PAÍS

Las universidades hondureñas han incrementado la oferta en ingeniería en sistemas, informática y telecomunicaciones, y existen programas de posgrado especializados. Sin embargo, la capacitación en DevOps, metodologías ágiles y certificaciones internacionales aún se limita. Este déficit de competencias digitales es uno de los principales retos para que el país aproveche el nearshoring.

Honduras enfrenta importantes desafíos relacionados con el capital humano, incluido el acceso limitado a una educación de calidad, las brechas de habilidades en la fuerza laboral y los elevados niveles de informalidad. El sistema educativo del país lucha contra la financiación, la infraestructura y la formación docente inadecuadas, lo que da como resultado bajos niveles de logro educativo y una falta de habilidades necesarias en la fuerza laboral. Esto obstaculiza la productividad y la innovación, limitando el crecimiento económico y las oportunidades de desarrollo. Además, una parte sustancial de la fuerza laboral opera en el sector informal, que hoy representa una parte significativa de la actividad económica del país, estimada en alrededor del 70 % del empleo total, sin acceso a protecciones sociales y oportunidades de capacitación, exacerbando las desigualdades y perpetuando la pobreza. Además, la fuga de cerebros sigue siendo una preocupación, ya que los trabajadores calificados a menudo buscan oportunidades en el extranjero debido a las perspectivas limitadas y la inestabilidad económica en el país. Abordar estos desafíos de capital humano requiere invertir en programas de educación y formación profesional, así como en políticas para promover el empleo formal y crear vías para el desarrollo y la retención de habilidades. Al priorizar el desarrollo del capital humano, Honduras puede liberar su potencial para el crecimiento sostenible y la prosperidad Find an expert (2024).

#### 2.2.5 LIMITACIONES Y RETOS NACIONALES

El país enfrenta falta de estandarización en procesos de desarrollo, una baja inversión en I+D ubicándose entre los niveles más bajos de la región (UNESCO, 2022), y una resistencia cultural al cambio tecnológico. Estas limitaciones se traducen en retrasos en proyectos de transformación digital y generan una brecha entre las expectativas del mercado y la capacidad de entrega de las empresas.

Las limitaciones en la organización también se dan por la resistencia a los cambios, ya que muchas empresas se mantienen en estructuras jerárquicamente rígidas y con una mentalidad de

control tradicional que choca con lo que es la filosofía ágil de autonomía. En el país, compañías como GBM y rubros como la banca hacen uso de metodologías ágiles como Scrum y Kanban. El factor financiero también juega un rol importante para la adopción, ya que para el manejo se necesita certificarse y actualmente son pocas o nulas las oportunidades de capacitación en Honduras.

### 2.2.6 INFRAESTRUCTURA TECNOLÓGICA

Aunque Honduras no cuenta con regiones propias de AWS, Azure o GCP, proveedores como Tigo Business han facilitado la adopción de servicios en la nube a través de acuerdos regionales. En 2025, el evento Unlocking Cloud Potential mostró cómo la alianza entre AWS y Tigo impulsa soluciones multicloud y de orquestación de contenedores (La Prensa, 2025). Sin embargo, la conectividad sigue siendo un desafío, la UIT reporta que la cobertura en zonas rurales se limita, lo que condiciona el despliegue de servicios basados en CI/CD.

Varias compañías en Honduras enfrentan retos en la conectividad, ya sea por las zonas donde están ubicadas o los costos que se requieren para financiar una infraestructura tecnológica robusta y empresarial. Las metodologías ágiles requieren herramientas en la nube como Jira, GitHub, Azure DevOps y Zoom. Sin embargo, los fallos de conectividad restringen su uso. Además, los costos de energía eléctrica y los cortes frecuentes afectan a los sectores industriales. Las organizaciones necesitan contar con equipos adecuados para cada uno de los integrantes del equipo de desarrolladores y testers, ya que no se pueden obtener resultados acertados; esto retrasa la madurez en prácticas ágiles y DevOps.

### 2.2.7 CONTEXTO EMPRESARIAL - NUEVA GENERACIÓN DE SOFTWARE (NGS)

#### 2.2.7.1 PERFIL DE LA EMPRESA

La Consultora Nueva Generación de Software (NGS) es una empresa hondureña dedicada al desarrollo de soluciones tecnológicas, integración de sistemas y consultoría.

#### 2.2.7.2 MISIÓN

Ser una empresa con liderazgo y con capacidad de proyección, que ofrezca productos y servicios de vanguardia que proporcionen valor agregado a nuestros clientes. Contando siempre con el compromiso de brindar transparencia y dedicación y, por sobre todas las cosas, logrando mantener los valores éticos y profesionales que siempre nos han caracterizado.

### 2.2.7.3 VISIÓN

Convertir a NGS en una de las 5 empresas de mayor importancia en Honduras en el área de desarrollo de software y consultorías informáticas, debido al nivel de sus operaciones a nivel nacional y el grado de satisfacción de nuestros clientes.

### 2.2.7.4 PRINCIPALES SERVICIOS

**Asesoría y consultoría:** A través de nuestro recurso humano, nuestros clientes obtienen asesoría y soporte especializado de productos y servicios, soporte de ingeniería en sistemas y software. NGS proporciona recursos especializados que desempeñan trabajos de consultoría, proporcionando al cliente una visión técnica de alto nivel, con el propósito de apoyar la implementación de nuevas tecnologías, estrategias y soluciones de software.

**Integración de sistemas:** Consiste en hacernos cargo de todas las fases requeridas en un proyecto completo de tecnologías de información IT. Desde el diseño hasta la implantación y puesta en marcha de los sistemas. Dentro de nuestras actividades principales:

1. Analizamos las necesidades reales de la empresa y ayudamos a alcanzar sus objetivos con el uso de tecnologías de vanguardia.
2. Proponemos soluciones de acuerdo con mejores prácticas y estándares de TI.
3. Diseñamos la solución de acuerdo con los recursos de la empresa.
4. Realizamos prototipos y modelos que faciliten el diseño de aplicaciones complejas.
5. Ejecutamos la construcción de las aplicaciones.
6. Brindamos servicio de soporte y mantenimiento de aplicaciones y bases de datos (preventivo y correctivo).
7. Efectuamos actividades de migración de datos, respaldo y recuperación de información.
8. Soluciones para Internet e Intranet.
9. Soluciones de Inteligencia de Negocios.

**Capacitación técnica especializada:** A fin de lograr una transferencia tecnológica

adecuada, NGS proporciona servicios específicos de capacitación para sus clientes, logrando que las soluciones implantadas logren ser sostenidas y gestionadas por recursos locales dentro de las organizaciones. Esto representa una ventaja para nuestros clientes y una de nuestras principales virtudes.

**Outsourcing del desarrollo de Software:** En la actualidad, el desarrollo de proyectos de TI ha generado la necesidad de contar con recursos especializados, lo cual demanda el crecimiento de las planillas de personal. El outsourcing del desarrollo de Software y los servicios relacionados con el mismo, constituye un enorme beneficio al reducir los costos operativos y permitir a la empresa enfocarse en sus procesos de negocios de forma eficiente.

#### 2.2.7.5 METODOLOGÍAS ACTUALES

NGS lleva a cabo sus proyectos de software aplicando metodología ágil, haciendo hincapié en técnicas que se basan en Kanban y Scrum. Estas técnicas han posibilitado que la institución opere de forma iterativa con entregas continuas e intermitentes, promoviendo el trabajo en equipo y la cooperación con los consumidores. No obstante, la aplicación todavía presenta restricciones en lo que respecta a la estandarización y a la incorporación de instrumentos de automatización.

#### 2.2.7.6 ESTADO ACTUAL DE LA ENTREGA CONTINUA

En NGS, la adopción de la entrega continua está en sus inicios y se nota por el uso parcial de entornos de validación como QA y preproducción, además de prácticas de integración continua (CI). Aunque el estado actual de la entrega continua en NGS muestra un progreso importante en CI, el despliegue continuo tiene una madurez baja, lo que constituye simultáneamente un obstáculo y una oportunidad. Al pasar a un modelo de DevOps más sólido, se podrán recortar los plazos de entrega, elevar la calidad del software y aumentar la confianza del cliente en los productos que se desarrollen.

#### 2.2.7.7 OPORTUNIDADES ESPECÍFICAS

1. Automatización de las pruebas y despliegues, lo que reducirá errores y tiempos de entrega.
2. Implementar flujos de Integración Continua (CI) y Entrega Continua (CD) para lograr ciclos de desarrollo más ágiles, estables y predecibles.
3. Incorporar monitoreo y retroalimentación continua, lo que permitirá detectar

problemas en etapas tempranas y mejorar la calidad de los productos en tiempo real.

4. Fortalecer el aseguramiento de la calidad (QA) con métricas objetivas que guíen la toma de decisiones.
5. Mejorar la colaboración entre áreas de desarrollo y operaciones, fomentando una cultura de trabajo transversal y compartida.

#### 2.2.7.8 IMPACTO ESPERADO DE LA PROPUESTA

**Reducción de los tiempos de entrega:** Realizando una automatización de pruebas y despliegues, la integración continua y la buena colaboración en los equipos, podremos tener un mejor control en los tiempos de entrega.

**Incremento en la calidad del software:** La calidad del software es fundamental, al realizar la integración correcta de las metodologías, se podrá tener una retroalimentación temprana y los despliegues controlados. Esto reduce los errores que se pueden presentar y mejorará la estabilidad de las entregas, asegurando el software con los requerimientos solicitados por el cliente.

**Mayor satisfacción del cliente:** Adoptar buenas prácticas en el ciclo de vida de un proyecto impulsa la satisfacción del cliente, ya que permitirá las entregas a tiempo y la mejor experiencia del usuario por la calidad del software.

**Optimizar recursos relacionados con costos por fallos:** La detección temprana de las fallas del software hace que se eviten retrabajos, lo que hace que disminuyan los costos por incidencias que se presenten en ambientes productivos. De esa forma se aprovecha mejor el tiempo, el presupuesto destinado para el proyecto y el equipo.

### 2.3 TEORÍA DE SUSTENTO

La teoría de sustento tiene como objetivo brindar los principios conceptuales que dirigen la investigación y otorgan coherencia a la propuesta presentada. La teoría de sustento, según Sampieri y Mendoza (2022), posibilita la creación de un marco conceptual firme que vincula los propósitos del trabajo con los principios y enfoques ya validados en la literatura, garantizando así que el estudio sea pertinente y riguroso.

### 2.3.1 TEORÍA DE LA CUARTA REVOLUCIÓN INDUSTRIAL

La cuarta Revolución Industrial se refiere al establecimiento de un sistema de producción flexible y eficiente mediante la convergencia del sistema de fabricación y las TIC (Tecnologías de la Información y la Comunicación). Nos enfrentamos a rápidos cambios a nivel mundial. Las TIC son fundamentales para la cuarta revolución, ya que se debe responder a los diferentes tipos de demanda de los clientes y a todos los cambios que sufre un producto en todo su proceso de fabricación, donde se aplican mejoras a los nuevos sistemas. DevOps es una forma de trabajar en el desarrollo de software que busca que los programadores y los equipos de operaciones se comuniquen y colaboren mejor. En el contexto de la Industria 4.0 y las fábricas inteligentes, la tecnología debe adaptarse rápidamente a los cambios y producir distintos tipos de productos de manera ágil. Esto hace necesario que los procesos de desarrollo y de manufactura también sean más flexibles y cooperativos.

La aplicación de prácticas conduce al éxito de muchas empresas, tenemos un ejemplo de éxito más destacado en China, Xiaomi, donde actualiza su software semanalmente y con esto es una empresa que se está conduciendo aceleradamente al éxito (Park & Huh, 2018). Las soluciones de la industria 4.0 buscan la digitalización de los procesos y servicios para hacerlos de una forma más rápida, eficiente y ordenada. DevOps permite facilitar esa transformación digital integrándolas en el desarrollo y la parte operacional, mejorando la comunicación, automatización (integración, entrega, pruebas y tener un monitoreo continuo del ciclo en todo su proceso) y la calidad del software sin generar costos no planificados. Este enfoque no solo apoya la agilidad en la producción, sino que también contribuye a que las empresas sobrevivan y prosperen en la era de la Cuarta Revolución Industrial al permitir operaciones más eficientes y mejores niveles de servicio (Suescún-Monsalve et al., 2021).

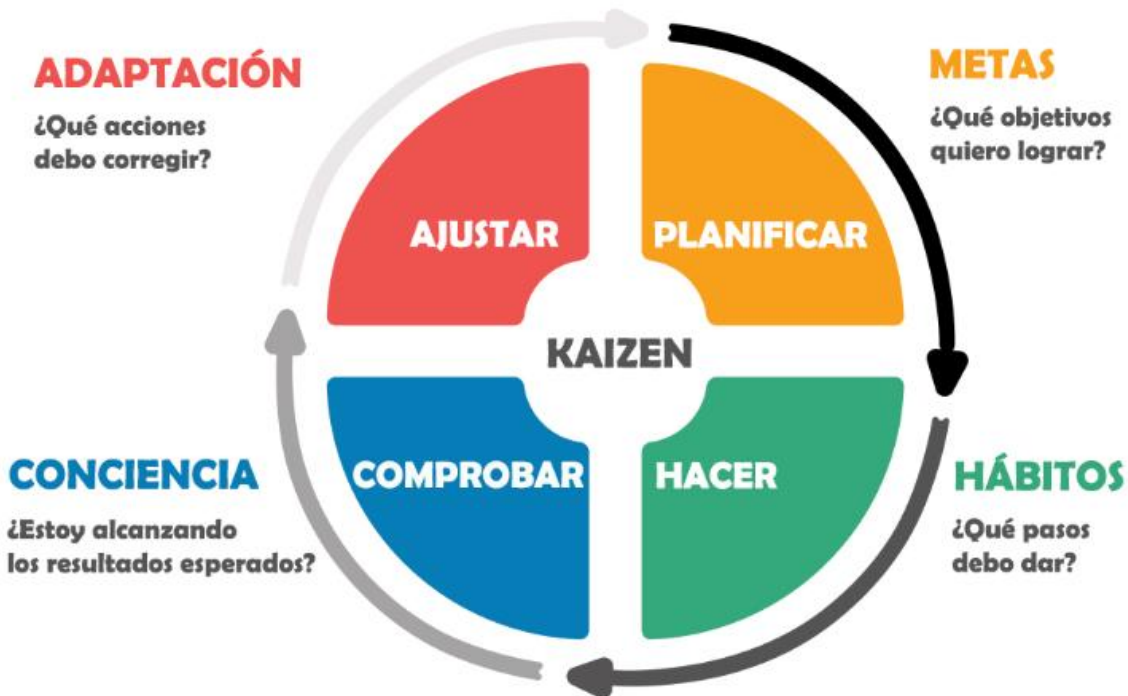
### 2.3.2 TEORÍA DE LA INNOVACIÓN DISRUPTIVA (CHRISTENSEN)

La teoría de la innovación disruptiva propuesta por Christensen (1997) plantea que las innovaciones disruptivas permiten a organizaciones con limitaciones superar a las incumbentes al ofrecer soluciones más rápidas y eficientes. En este marco, DevOps se considera una innovación de proceso que rompe con el modelo tradicional en el que desarrollo y operaciones trabajaban de manera aislada, habilitando despliegues frecuentes y confiables, y generando ventajas competitivas sostenibles. DevOps rompe con el modelo tradicional de separación entre desarrollo y la operatividad, ya que hace la integración de los procesos entre automatizar y entrega continúa

ofreciendo a las empresas las posibilidades de competir con rapidez, calidad y reducir costos, es por ello por lo que las organizaciones que adoptan DevOps pueden tener grandes ventajas ante aquellas que siguen aferrándose a los modelos tradicionales.

### 2.3.3 TEORÍA DE LA MEJORA CONTINUA (KAIZEN / CICLO PDCA)

**Figura 3** Teoría de la mejora Continua (Kaizen/ Ciclo PDCA)



Fuente: Tomado de Kaoru Ishikawa Institute (2023).

La filosofía Kaizen, junto con el ciclo PDCA de Deming, fundamenta la idea de Iterar en ciclos cortos, medir resultados y ajustar procesos continuamente (Imai, 1986). En DevOps, esto se refleja en la integración y entrega continua (CI/CD), donde se planifica, se construye y despliega software, se miden métricas como la DORA (lead time, frecuencia de despliegue, MTTR, tasa de fallos de cambio) y se retroalimenta para la mejora continua. Este enfoque plantea que los cambios no se logran a través de transformaciones radicales inmediatas, sino mediante la implementación progresiva de mejoras pequeñas y continuas que, en conjunto, generan un impacto significativo.

## 2.4 METODOLOGÍAS

### 2.4.1 SCRUM COMO MARCO DE TRABAJO ÁGIL

El desarrollo de Software es una tarea compleja que ha dado lugar a diversas metodologías. Las tradicionales se enfocan en el control estricto del proceso, definiendo actividades, artefactos y herramientas, lo que ha sido útil en muchos proyectos, pero en otros ha generado rigidez y complejidad excesiva. Como alternativa, surgen las metodologías ágiles, que priorizan al individuo, la colaboración con el cliente y el desarrollo incremental mediante iteraciones cortas. Este enfoque ha demostrado ser especialmente eficaz en proyectos con requisitos cambiantes y con exigencia de rapidez sin sacrificar calidad. Actualmente, las metodologías ágiles están transformando la manera de producir software, aunque generan debate entre quienes las defienden y quienes siguen confiando en los enfoques tradicionales (Canós, Letelier, & Penadés, s. f.).

Scrum no es solo un conjunto de prácticas, sino un marco de trabajo que brinda visibilidad, permite inspeccionar y adaptar continuamente, y expone las disfunciones que afectan al equipo y al Product Owner. No es su propósito solucionar directamente los problemas, sino hacerlos visibles para que se puedan tratar a través de ciclos breves y mejoras constantes. Un ejemplo es cuando un equipo no cumple lo comprometido en su primer sprint, lejos de ser un fracaso. Es una oportunidad para aprender a estimar mejor y ser más realistas. Este mecanismo de “exponer y corregir” es la base de los beneficios que Scrum ofrece. Sin embargo, existen fallos comunes como cambiar Scrum en lugar de cambiar la forma de trabajar (ej. alargar Sprints para evitar mejorar la estimación). Creer que Scrum prohíbe prácticas no mencionadas explícitamente (ej. estrategia a largo plazo del Product Owner o mentoría técnica entre ingenieros). Imponer Scrum desde la dirección, en vez de permitir que el equipo lo adopte de forma autónoma y con apoyo de un Scrum Master experimentado. Finalmente, aunque el primer sprint suele ser desafiante, los equipos suelen reconocer rápidamente los beneficios y concluyen que, aunque exigente, Scrum es mucho mejor que los métodos anteriores.

**Figura 4** Elementos principales en la creación de un producto



Fuente: Adaptada de metodología Scrum (Trigás Gallego, 2012). Universitat Oberta de Catalunya.

Scrum es una metodología ágil, empírica, iterativa e incremental diseñada para gestionar proyectos de software en entornos complejos y cambiantes. Su propósito es mejorar la comunicación, reducir la incertidumbre y optimizar la entrega de valor en cada iteración.

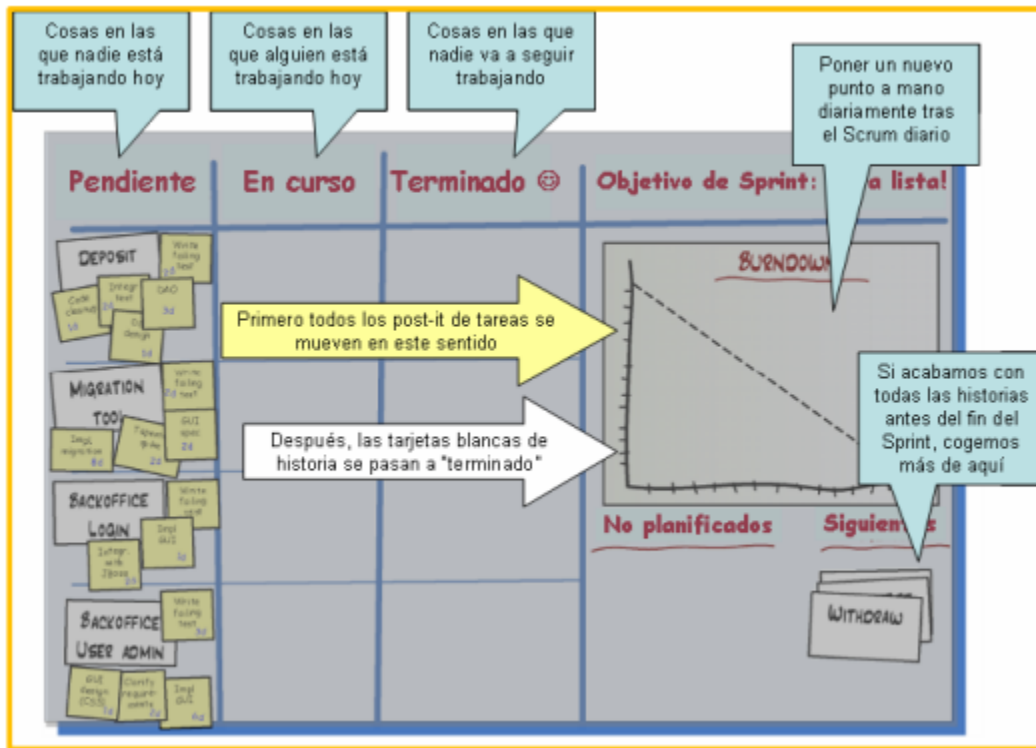
Se basa en la idea de que el desarrollo de software es inherentemente impredecible, por lo que propone trabajar en ciclos cortos llamados sprints, usualmente de un mes, donde se construye un incremento funcional del producto.

En Scrum, los requerimientos se gestionan mediante un Product Backlog, una lista priorizada de funcionalidades, tareas o historias de usuario que se define junto con los stakeholders. En cada sprint, se selecciona un subconjunto de este backlog (Sprint Backlog) que será trabajado por el equipo (Universidad Nacional de Trujillo, s. f.).

El cliente podría utilizar los resultados más significativos del proyecto antes de que esté finalizado por completo. El cliente establece sus expectativas indicando el valor que le aporta cada requisito. Se reducen los riesgos porque, al momento de tener sprints cortos, se pueden mitigar los cambios imprevistos. Se pueden hacer reducciones y predicciones de tiempos, al llevar diariamente una reunión diaria y realizar un Sprint Burnout Chart para la visibilidad del proyecto. Mayor productividad porque el equipo sabe diariamente qué es lo que va a hacer cada miembro del equipo.

Mayor tolerancia al cambio.

**Figura 5** Scrum taskboard



Fuente: Tomada de Metodología Scrum (Trigás Gallego, 2012). Universitat Oberta de Catalunya.

Como toda metodología, Scrum presenta tanto puntos fuertes como limitaciones o restricciones que deben considerarse al momento de su implementación:

#### 2.4.1.1 FORTALEZAS DE SCRUM

1. Adaptabilidad y flexibilidad: Su método iterativo, basado en ciclos de retroalimentación y sprints breves, posibilita una rápida reacción a las modificaciones en los requisitos, lo cual es más complicado con métodos tradicionales.
2. Suministro y retroalimentación anticipada: Permite la entrega ininterrumpida de software que funciona, lo cual garantiza una validación permanente por parte del cliente y un mejor acoplamiento con sus expectativas.
3. Claridad: Los artefactos visibles y los encuentros regulares (diarios, revisiones de sprint) aumentan la visibilidad del progreso y los obstáculos para las partes interesadas y para el equipo.
4. Colaboración y fortalecimiento del equipo: Promueve que cada miembro participe

activamente y se autoorganice, lo cual mejora la calidad de las decisiones tomadas y la motivación.

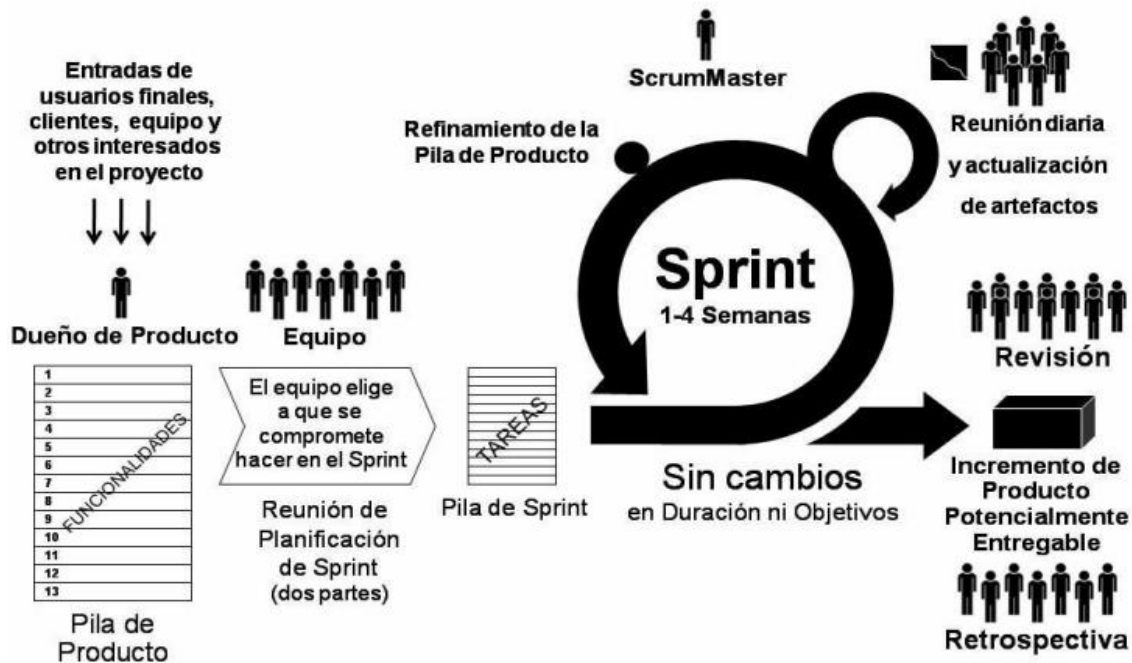
#### 2.4.1.2 LIMITACIONES DE SCRUM

1. Gran dependencia del compromiso: Si no hay una colaboración, confianza y compromiso total de todos los miembros del equipo, su eficacia se ve afectada.
2. Requerimiento de un Scrum Master con experiencia: Un rol que se ejerce de manera inadecuada puede perjudicar el adecuado uso de las prácticas de Scrum.
3. Riesgo de que el alcance se desplace: Si no se gestiona con disciplina, la flexibilidad puede llevar a que los objetivos del proyecto se amplíen de manera incontrolada.

#### 2.4.1.3 ROLES

Al aplicar las metodologías Scrum se definen las siguientes responsabilidades: No existe jefe de proyecto, ya que se distribuye en los roles de un equipo Scrum: el cliente o Product Owner, Scrum máster o facilitador y el resto del equipo. El objetivo de Scrum es maximizar la realimentación sobre el desarrollo, en el cual corrige problemas y mitiga riesgos de manera temprana.

**Figura 6** Roles, artefactos y eventos principales



Fuente: Obtenida de Leo Antoli et al., s. f., p. 5.

**Tabla 6** Roles de la metodología Scrum

Roles	Responsabilidades Principales
<b>Cliente (Product Owner)</b>	Representa a los usuarios finales y a los interesados.
	Establece y transmite los propósitos del producto o proyecto.
	Administra y da prioridad al Product Backlog.
	De acuerdo con el valor y el contexto (mercado, competencia, demandas), planifica y reordena las entregas.
	Toma parte en la planificación de iteraciones, así como en la revisión y demostración.
	Está a disposición durante la iteración y responde preguntas.
	Válida y aprueba los entregables.
<b>Facilitador (Scrum Máster)</b>	Asegura que Scrum se aplique correctamente.
	Organiza reuniones (planificación, diarias, demostración y retrospectiva).
	Elimina obstáculos que impiden que el equipo trabaje.
	Protege al equipo de interrupciones externas.
	Fomenta la autogestión y la mejora continua.
	Asegura la calidad del proceso y que se cumpla el compromiso asumido.
<b>Equipo (Team)</b>	Grupo multidisciplinario y autogestionado (de 5 a 9 integrantes).
	Escoge los requisitos que se van a desarrollar en cada iteración.
	Calcula el esfuerzo y la complejidad de los requisitos.
	Establezca la manera en la que llevará a cabo el trabajo.
	Comparte la responsabilidad en cuanto a la calidad del producto.
	Proporciona mejoras de valor al cliente en cada iteración.

Nota: Elaboración propia.

#### 2.4.1.4 CICLO DE VIDA DE SCRUM

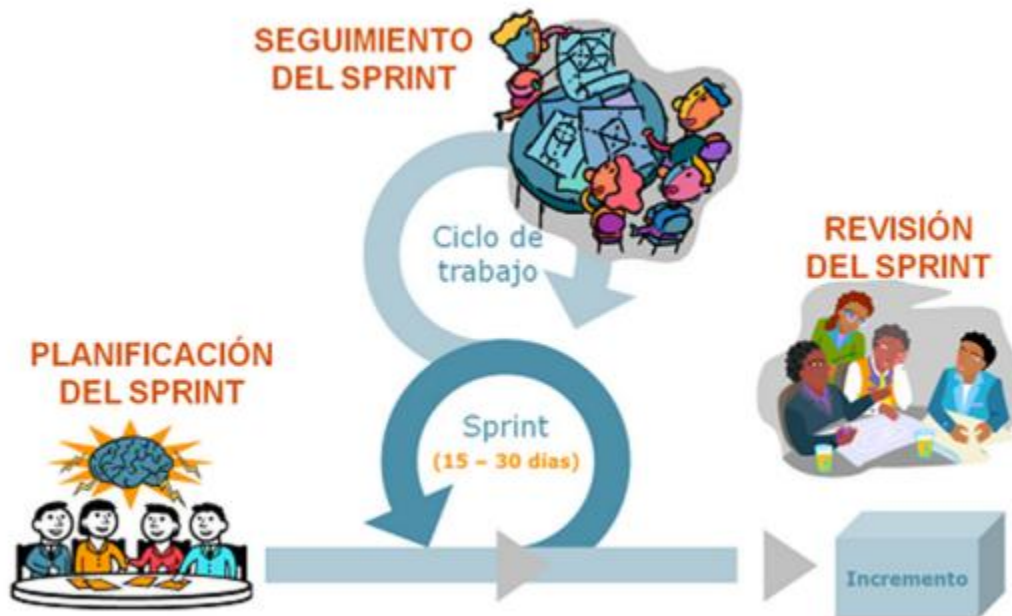
Dado que la mayoría de las organizaciones depende de sus aplicaciones para funcionar de manera eficiente, el desarrollo de software con criterios de seguridad se ha convertido en un aspecto crucial para ellas. Es por ello que es esencial adoptar metodologías de desarrollo seguras que se apliquen en todas las etapas del ciclo de vida, incluyendo la recopilación de requisitos, el diseño, la codificación y las pruebas.

**Tabla 7** *Fases del ciclo de vida de Scrum*

<b>Fase</b>	<b>Propósito</b>	<b>Actividades</b>	<b>Resultado</b>
<b>Pre-juego: Planeamiento</b>	Establecer la visión del proyecto, definir expectativas y asegurar la financiación.	Redacción de la visión, elaboración del presupuesto, creación del Product Backlog inicial con ítems estimados, diseño exploratorio, prototipos y definición de la arquitectura de alto nivel.	Plan de alto nivel que orienta el proyecto y marca la dirección inicial.
<b>Pre-juego: Montaje (Staging)</b>	Refinar y priorizar los requerimientos para preparar la primera iteración.	Planificación detallada, diseño exploratorio y ajustes a los prototipos	Sprint Backlog inicial con las tareas priorizadas para el primer ciclo
<b>Juego o desarrollo</b>	Implementar un sistema funcional y listo para entregar, mediante iteraciones de 30 días (Sprints).	Planeación de cada Sprint, construcción del incremento del producto, definición y refinamiento del Sprint Backlog y Daily Scrum para seguimiento y resolución de impedimentos.	Incremento del producto terminado y funcional al final de cada Sprint.
<b>Pos-Juego: Liberación</b>	Realizar el despliegue operacional del producto.	Documentación, entrenamiento de usuarios, mercadeo, soporte y ventas.	Producto entregado y puesto en operación para los usuarios finales.

Nota: Elaboración propia.

**Figura 7** Ciclo de carrera o de vida (Sprint) de Scrum



Fuente: Obtenida de Pérez, s.f., p. 19.

#### 2.4.1.5 RETROSPECTIVA DEL SPRINT EN LA METODOLOGÍA SCRUM

La revisión del Sprint se centra en inspeccionar y adaptar el producto, mientras que la retrospectiva del Sprint se enfoca en revisar y modificar el proceso. Aunque algunos equipos la omiten, es clave porque permite identificar áreas de mejora y convertirlas en resultados concretos. En la retrospectiva, el equipo analiza qué funcionó bien y qué debe mejorar. Se suelen usar dos columnas (“Qué fue bien” y “Qué se podría mejorar”) y, tras identificar los temas más recurrentes, se buscan causas y se acuerdan pocos cambios a aplicar en el siguiente Sprint.

El Scrum Master suele facilitar la reunión, aunque también puede hacerlo alguien externo para mayor objetividad. Incluso se recomienda que Scrum Masters faciliten retrospectivas de otros equipos, fomentando el aprendizaje cruzado (Scrum Primer, Deemer, Benefield, Larman & Vodde, s. f., p. 18).

#### 2.4.2 DEVOPS COMO MODELO DE INTEGRACIÓN Y ENTREGA CONTINUA

Los principios de DevOps no constituyen una especificación rígida, un estándar formal ni un conjunto de leyes o normativas; más bien, representan un marco de referencia flexible. Tal como señala un artículo de GitLab, lo esencial de DevOps radica en sus principios, los cuales impulsan la mejora de las prácticas de desarrollo de software dentro de una organización. Entre

los más relevantes se destacan: la automatización del ciclo de vida del software, la comunicación y colaboración efectiva entre equipos, la mejora continua, la reducción de desperdicios y, finalmente, el enfoque en las necesidades de los usuarios, respaldado por ciclos cortos de retroalimentación (GitLab, 2022).

#### 2.4.2.1 PRINCIPIOS CULTURALES DE DEVOPS

**Colaboración:** DevOps se basa en la colaboración continua de equipos multidisciplinarios, que comparten responsabilidad, comunicación constante y trabajo conjunto durante todo el ciclo de vida del software, desde la idea hasta la entrega final.

**Automatización:** Una práctica clave de DevOps es la automatización del ciclo de vida del software, lo que libera tiempo para que los equipos se enfoquen en tareas de mayor valor, como crear nuevas funcionalidades y corregir errores. La automatización es un elemento clave de CI/CD y ayuda a reducir los errores humanos y aumentar la productividad del equipo.

**Mejora continua:** La mejora continua se convirtió en una parte integral de las prácticas ágiles. Se trata de alentar la experimentación, minimizar el desperdicio y maximizar la velocidad, el costo y la facilidad de entrega. Además, está muy relacionada con la entrega continua, ya que permite a los equipos DevOps realizar actualizaciones continuas para mejorar la eficiencia y la calidad del software. DevOps se basa en la colaboración continua de equipos multidisciplinarios, que comparten responsabilidad, comunicación constante y trabajo conjunto durante todo el ciclo de vida del software, desde la idea hasta la entrega final.

**Acción centrada en el cliente:** Los equipos DevOps emplean ciclos cortos de retroalimentación y monitoreo en tiempo real para responder rápidamente a los usuarios, obteniendo visibilidad inmediata de su interacción con las aplicaciones y aplicando mejoras continuas.

**Responsabilidad de extremo a extremo:** La responsabilidad de extremo a extremo implica que todo el equipo comparte éxito y fracaso como una sola entidad, fomentando propiedad, motivación y transparencia, al tiempo que mantiene límites claros y una rendición de cuentas sana.

**Crear con el objetivo en mente:** Este principio destaca que los equipos DevOps deben entender las necesidades reales de los clientes y evitar desarrollar en aislamiento, adoptando una visión integral del producto desde su creación hasta su implementación.

**Tabla 8** *Procesos y Prácticas de DevOps*

<b>Procesos y Prácticas</b>	<b>Descripción</b>
<b>Integración Continua</b>	La integración continua (CI) consiste en validar de forma frecuente los cambios al código, lo que incrementa la confianza del equipo en la gestión de modificaciones y asegura mayor calidad en el desarrollo.
<b>Entrega Continua</b>	La entrega continua (CD) es la práctica de poner cambios en entornos de prueba y luego en producción en cortos plazos, permitiendo a las partes interesadas y a los usuarios finales percibir rápidamente el valor agregado del software.
<b>Despliegue Continuo</b>	El despliegue continuo automatiza por completo la entrega, llevando cada cambio validado directamente a producción sin intervención humana, salvo que falle una prueba.
<b>Pruebas Continuas</b>	Las pruebas continuas incluyen verificaciones automáticas en la integración continua, para que cada nueva función se revise, haciendo especial hincapié en los procesos clave del negocio y en la seguridad de los cambios.
<b>Monitoreo y Alertas</b>	El monitoreo y las alertas dan visibilidad al estado de los sistemas, ayudan a comprender tendencias y el impacto de los cambios, y avisan cuando las métricas se desvían de lo normal para ayudar a encontrar la causa (DigitalOcean, 2017).

Nota: Elaboración propia adaptada de Diseño de guía de adaptación de prácticas DevOps para empresas con procesos de ingeniería de software.

### 2.4.3 ITIL V4 COMO MARCO DE GESTIÓN DE SERVICIOS TECNOLÓGICOS

ITIL v4 es un marco de trabajo que proporciona directrices para la gestión de servicios de TI, enfocándose en la entrega de valor al cliente y la mejora continua. Esta versión combina ideas de metodologías ágiles, Lean y DevOps para ajustarse a las dinámicas actuales del mundo empresarial. Fomenta entregas en ciclos cortos, automatización de procesos, trabajo en equipo y un enfoque continuo en crear valor y mejorar la eficiencia. (Axelos, 2020)

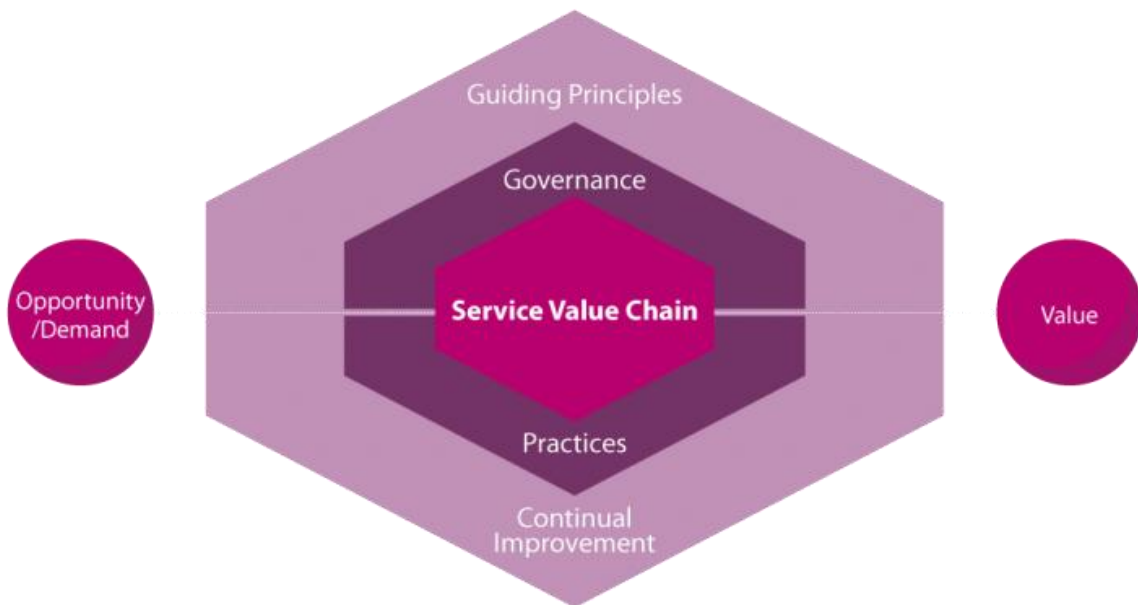
ITIL se compone de procesos definidos y documentados que abarcan desde la planificación hasta la entrega y mejora continua de servicios de TI. A pesar de no existir un estándar internacional formal único para ITSM, ITIL se considera prácticamente el estándar de facto global para alinear TI con la estrategia del negocio. ITIL 4 amplía las versiones anteriores al proporcionar un modelo operativo práctico y flexible que integra prácticas ágiles y modernas. Por ejemplo, según Behr et al. (2021), “ITIL 4 también ofrece una visión holística de extremo a extremo que integra marcos como Lean, Agile y DevOps”.

ITIL 4 es un marco de gestión de servicios de TI orientado a la creación de valor, que

sistematiza procesos de TI mediante un ciclo de servicio integrado en un contexto empresarial y tecnológico moderno.

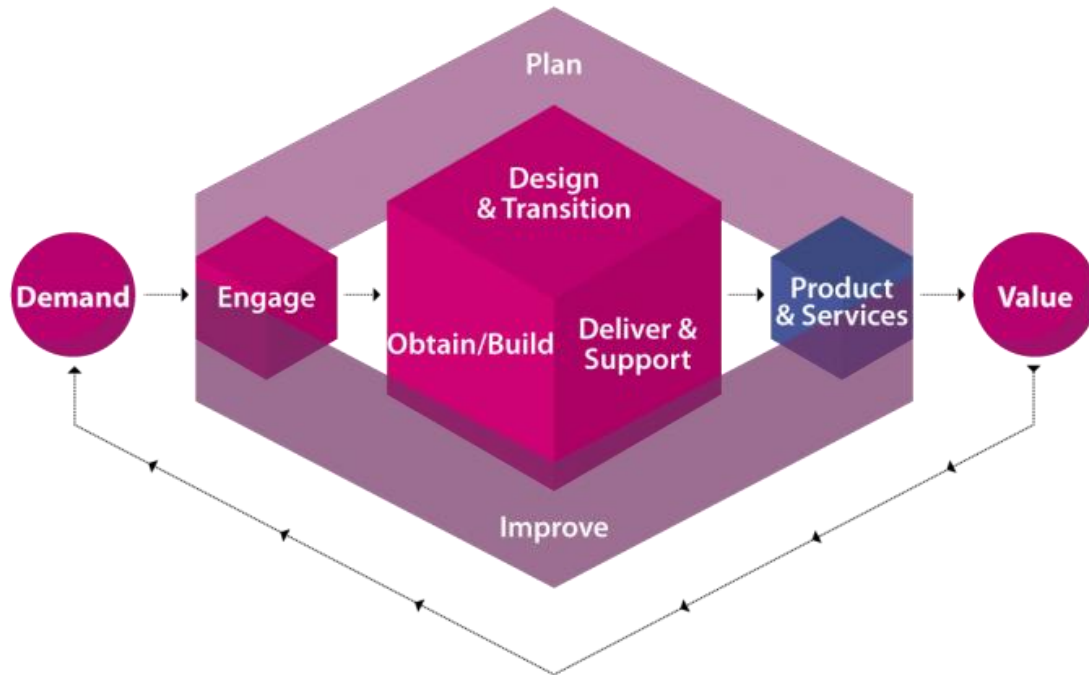
Uno de los pilares fundamentales de ITIL V4 es el Sistema de Valor del Servicio (SVS), el cual integra diversos componentes y actividades de la organización para facilitar la cocreación de valor a través de servicios habilitados por TI. Este sistema promueve una visión holística que abarca desde los principios rectores hasta la mejora continua, asegurando que cada elemento de la organización contribuya al valor final entregado al cliente. (Shahi, 2023)

**Figura 8** Sistema de Valor de Servicio (Service Value System) de ITIL 4, con sus cinco componentes clave: principios rectores, gobernanza, cadena de valor del servicio, prácticas de gestión y mejora continua.



Fuente: Adaptado de Componentes de ITIL 4, por ITIL México, s.f.

**Figura 9** Cadena de Valor del Servicio (Service Value Chain) en el Sistema de Valor del Servicio (SVS) de ITIL 4.



Fuente: Adaptado de Componentes de ITIL 4, por ITIL México, s.f.

## 2.5 ANTECEDENTES DE LAS METODOLOGÍAS

### 2.5.1 EXPERIENCIAS PREVIAS CON SCRUM

Scrum es uno de los marcos más influyentes dentro del universo ágil. Su nacimiento en los años 90 coincidió con otras propuestas como Extreme Programming (XP) y Feature Driven Development (FDD), que compartían la misma crítica, los modelos tradicionales como Cascada eran rígidos, burocráticos y lentos para proyectos de software con alta incertidumbre. Takeuchi y Nonaka (1986) aportaron un antecedente clave para todas las metodologías ágiles al introducir el concepto de equipos autoorganizados, multifuncionales y flexibles, inspirado en el rugby. Esta idea dio sustento a Scrum y permeó a otras metodologías ágiles que priorizan equipos autónomos y aprendizaje continuo (Schwaber & Sutherland, 2020).

En 1996, Jeff Sutherland y Ken Schwaber presentaron las prácticas que se usaban como proceso formal para el desarrollo de software y que pasarían a incluirse en la lista de Agile

Alliance. Scrum se adecua para aquellas empresas en las que el desarrollo de los productos se realiza en entornos que se caracterizan por tener: Incertidumbre, autoorganización, control moderado y transmisión del conocimiento.

Scrum, como una forma de gestión ágil de proyectos, se caracteriza por su enfoque iterativo e incremental para el desarrollo de productos, donde el trabajo se divide en 'sprints' o ciclos cortos de trabajo. Con el tiempo, la comprensión y la aplicación de Scrum han evolucionado, con una variedad de interpretaciones e implementaciones por parte de diferentes equipos y organizaciones de todo el mundo. A pesar de esto, los valores y principios importantes, como se explica en la Guía oficial de Scrum, se han mantenido bastante estables y, en general, se acepta que seguirlos es necesario para una verdadera práctica de Scrum.

### 2.5.2 EXPERIENCIAS PREVIAS CON DEVOPS

DevOps ya no es un concepto considerado como “nuevo”, puesto que desde su concepción original ya han pasado más de 10 años. Sin embargo, nuestra historia empieza en 2008, donde Patrick Debois<sup>1</sup> presentó por primera vez el concepto de infraestructura ágil en una conferencia en Toronto llamada Agile 2008. En 2009, Patrick organizó la primera conferencia relacionada con DevOps con el nombre de DevOpsDays<sup>2</sup>, donde el objetivo principal era traer a profesionales de desarrollo de software y operaciones de TI a trabajar en conjunto.

Inicialmente, el tema de la conferencia iba a ser Administración de Sistemas Ágil (“Agile System Administration” en inglés), pero para Patrick esto era un título muy largo, así que como el enfoque de la conferencia sería orientado a Desarrollo y Operaciones trabajando en conjunto, combinó estas dos palabras obteniendo entonces DevOps. Nunca hubo una gran intención o propósito detrás de dicho nombre, solo se quería un nombre más corto.

De acuerdo con el vídeo The Short History of DevOps, luego de la primera instancia de DevOpsDays, las personas que participaron siguieron conversando sobre este tema en diferentes lugares de la Internet, principalmente en Twitter. Una de las razones del rápido esparcimiento de la palabra DevOps se debió a que en Twitter las discusiones relacionadas con la conferencia fueron etiquetadas con DevOpsDays. Sin embargo, debido a las limitantes de caracteres de Twitter por cada tuit en ese entonces, el hashtag evolucionó a ser simplemente #DevOps (Edwards, 2012).

Tras DevOpsDays, la idea de DevOps se transformó rápidamente en un movimiento emergente, impulsado en gran parte por la predicción de Cameron Haight de Gartner en 2011,

quien afirmó que para 2015 sería adoptado por el 20% de las organizaciones Global 2000. Aunque no se puede asegurar que su análisis fuera el detonante principal, a partir de esa fecha grandes empresas comenzaron a interesarse y adoptar DevOps (Haight, 2010). Desde 2011, DevOps ha representado una transformación en la ingeniería de software y operaciones, aportando beneficios a toda la organización y consolidándose como un viaje de mejora continua orientado al rendimiento, la eficiencia y la entrega constante de valor (Mezak, 2018).

### 2.5.3 EXPERIENCIAS PREVIAS CON ITIL 4

El marco ITIL (Information Technology Infrastructure Library) surgió en los años ochenta en el Reino Unido con el propósito de estandarizar la entrega eficiente y efectiva de servicios de TI. Originalmente promovido por la Oficina de Comercio Gubernamental británica (OGC), ITIL buscaba garantizar servicios informáticos de alta calidad, independientemente del proveedor. Con el tiempo, ITIL se convirtió en el estándar de facto para la gestión de servicios de TI a nivel mundial. De hecho, “en todo el mundo y para cualquier organización, ITIL se ha convertido en un estándar de facto [de la gestión de servicios TI]”. Desde la perspectiva del cliente y del negocio, ITIL ofrece buenas prácticas que se adaptan a las necesidades de cualquier empresa, permitiendo un mejor control, operación y administración de los recursos de TI.

La evolución de ITIL se ha materializado en varias versiones. En 2007 se publicó la versión ITIL v3, introduciendo un ciclo de vida de cinco fases (Estrategia, Diseño, Transición, Operación y Mejora Continua) documentado en cinco libros guía. Posteriormente, en 2019 se lanzó ITIL 4, una versión significativamente más flexible y orientada al valor del servicio. Esta nueva edición integró prácticas modernas y enfoques ágiles, como lo describen Sambrano et al., “ITIL v4 integrando prácticas modernas para mejorar la gestión de servicios”. ITIL 4 se diferencia de las versiones anteriores en su enfoque en la creación de valor y en la adaptabilidad a entornos dinámicos. Su estructura conceptual gira en torno al *Sistema de Valor del Servicio (SVS)*, que incluye la *Cadena de Valor del Servicio (SVC)* un modelo operativo de seis actividades clave (Planificar, Mejorar, Involucrar, Diseñar y Transición, Obtener/Construir, Entregar y Soportar) y las cuatro dimensiones de gestión de servicios (organizaciones y personas; información y tecnología; socios y proveedores; flujos de valor y procesos).

ITIL 4 incorpora además siete principios guía para orientar su implementación: enfocarse en el valor, comenzar donde se está, progresar iterativamente con retroalimentación, colaborar y

promover la visibilidad, pensar y trabajar de forma integral, mantenerlo simple y práctico, y optimizar y automatizar. En palabras de Williams (2021), “los siete principios de ITIL V4 guían a las organizaciones en su viaje hacia la mejora continua”. Estas directrices facilitan la adopción de ITIL en entornos diversos. Notablemente, ITIL v4 se diseñó para complementar metodologías ágiles y DevOps. De hecho, la literatura destaca que ITIL 4 se alinea estrechamente con prácticas ágiles y DevOps, permitiendo a las organizaciones ser más ágiles y adaptativas en contextos dinámicos. Por ejemplo, la Figura 1 muestra la evolución histórica del movimiento DevOps, cuya cultura contribuye al entorno en el que se insertan las buenas prácticas de ITIL.

**Tabla 9** Línea de tiempo de la evolución de ITIL

<b>Año</b>	<b>Versión / Evento</b>	<b>Descripción</b>
<b>1980 s</b>	Nacimiento de ITIL	Creado en el Reino Unido por la Oficina de Comercio Gubernamental (OGC) con el objetivo de estandarizar la entrega eficiente de servicios de TI y garantizar alta calidad en los servicios informáticos, independientemente del proveedor.
<b>2007</b>	ITIL v3	Introduce un enfoque basado en el ciclo de vida del servicio, estructurado en cinco fases: Estrategia, diseño, transición, operación y mejora continua, documentado en cinco libros guía.
<b>2011</b>	Actualización ITIL v3 (2011 Edition)	Se realizan mejoras editoriales y clarificaciones en los procesos, consolidando la v3 como estándar de facto para ITSM a nivel mundial.
<b>2019</b>	ITIL 4	Publicación de una versión más flexible, ágil y centrada en el valor del servicio. Integró prácticas modernas y enfoques ágiles. Introduce el Sistema de Valor del Servicio (SVS), la Cadena de Valor del Servicio (SVC) y las Cuatro Dimensiones de gestión de servicios.

<b>2019-2021</b>	Consolidación de ITIL 4	Inclusión de los 7 principios Guías que orientan la implementación: enfoque en el valor, comenzar donde se está, progresar iterativamente, colaboración y visibilidad, pensar y trabajar integralmente, simplicidad y automatización.
<b>Actualidad</b>	Complementariedad con Ágil y DevOps	ITIL 4 se alinea con prácticas ágiles y DevOps, permitiendo mayor agilidad, adaptabilidad y colaboración en entornos dinámicos y altamente cambiantes.

Nota: Elaboración propia con base en Sambrano et al. (2019), Williams (2021) y AXELOS (2019).

En este contexto, ITIL 4 enfatiza la colaboración, la automatización y la simplicidad, valores compartidos con DevOps y metodologías Lean. Su enfoque integral pretende alinear la entrega tecnológica con las necesidades del negocio. Como señala AXELOS (2019), ITIL “promueve un modelo de servicio que alinea de manera más integral la entrega de tecnología con las necesidades del negocio”. Esto es especialmente útil para organizaciones modernas, en las cuales las TI deben adaptarse continuamente a cambios de mercado y de la propia tecnología.

A nivel global, ITIL es ampliamente adoptado como la guía de referencia para la gestión de servicios. Ya desde 2012 se reportaba que “ITIL se ha convertido en un estándar de facto”, y la comunidad de TI lo reconoce como el modelo más consolidado para definir roles, funciones y responsabilidades en servicio. Esta difusión internacional incluye múltiples industrias (bancaria, salud, gobiernos, etc.) donde la alineación de TI con los objetivos de negocio resulta crítica.

En América Latina y Centroamérica también existen casos de éxito que muestran los beneficios de ITIL 4. Por ejemplo, estudios en empresas e instituciones latinoamericanas indican mejoras significativas tras su adopción. Huarcaya (2022) reportó que en una organización la tasa de resolución de incidentes aumentó del 85,4 % al 93,73 %, y la eficacia de cumplimiento de SLA pasó del 82,6 % al 93,93 %, mientras el tiempo medio de resolución se redujo en un 15,12 %. De manera similar, Táez (2021) registró en un municipio de Ecuador una mejora del 25,08 % en los tiempos de respuesta a incidentes tras implementar métricas basadas en ITIL 4. Estos resultados indican una mejora operativa y de satisfacción del cliente gracias a las prácticas de ITIL V4. Otros casos en Perú y Bolivia documentaron reducciones drásticas en tiempos de solución y aumentos en la eficiencia al adoptar la biblioteca de ITIL 4.

En Centroamérica, la adopción de ITIL V4 es también relevante. Un estudio reciente sobre la Mesa de Servicio del Banco Davivienda Honduras (2024) subraya que optimizar esta área bajo los estándares de ITIL V4 es crítico en países en desarrollo. En dicho contexto bancario, ITIL facilita alinear la entrega de tecnología con demandas del mercado y regulaciones locales. Por ejemplo, se reconoce que ITIL no solo mejora la eficiencia de los servicios, sino que también ayuda a las empresas a “comprender y cumplir con la legislación pertinente” relacionada con datos y seguridad. En Honduras, donde las regulaciones de protección de datos y ciberseguridad son cada vez más estrictas, ITIL V4 ofrece pautas que mitigan riesgos regulatorios al asegurar el cumplimiento de requisitos legales. Además, se enfatiza la adaptación cultural local, según Van Bon (2019), la “adaptación cultural es crucial para el éxito de cualquier optimización de gestión de servicios de TI”.

Finalmente, la experiencia regional apunta a que ITIL 4 contribuye a resultados estratégicos. El mencionado caso de Davivienda Honduras concluye que la adopción de ITIL permite alinear mejor las operaciones de TI con los objetivos del negocio, generando una ventaja competitiva en el mercado local y centroamericano. En suma, a nivel internacional y regional, los antecedentes muestran que ITIL 4, con su enfoque de valor y buenas prácticas probadas, es un modelo eficaz para optimizar la gestión de servicios de TI.

## **2.6 ANÁLISIS CRÍTICO DE LAS METODOLOGÍAS**

### **2.6.1 APLICABILIDAD DE SCRUM**

La metodología Scrum no es un conjunto de prácticas rígidas, sino un marco de trabajo que brinda visibilidad, permite inspeccionar y adaptar, y hace evidentes las debilidades o disfunciones en los equipos. Scrum no resuelve los problemas directamente, pero los vuelve visibles para que se puedan resolver a tiempo durante el proceso. Un error en un Sprint no debe verse como un fracaso, sino como un aprendizaje para comprometerse de manera más realista.

**Figura 10** *Aplicabilidad de la metodología Scrum*



Fuente: Elaboración propia tomada de Metodología Scrum (Trigás Gallego, 2012). Universitat Oberta de Catalunya.

### 2.6.1.1 REUNIONES EN EL SPRINT

Durante todo el Sprint se celebran 3 reuniones, las que dan una retroalimentación del proceso:

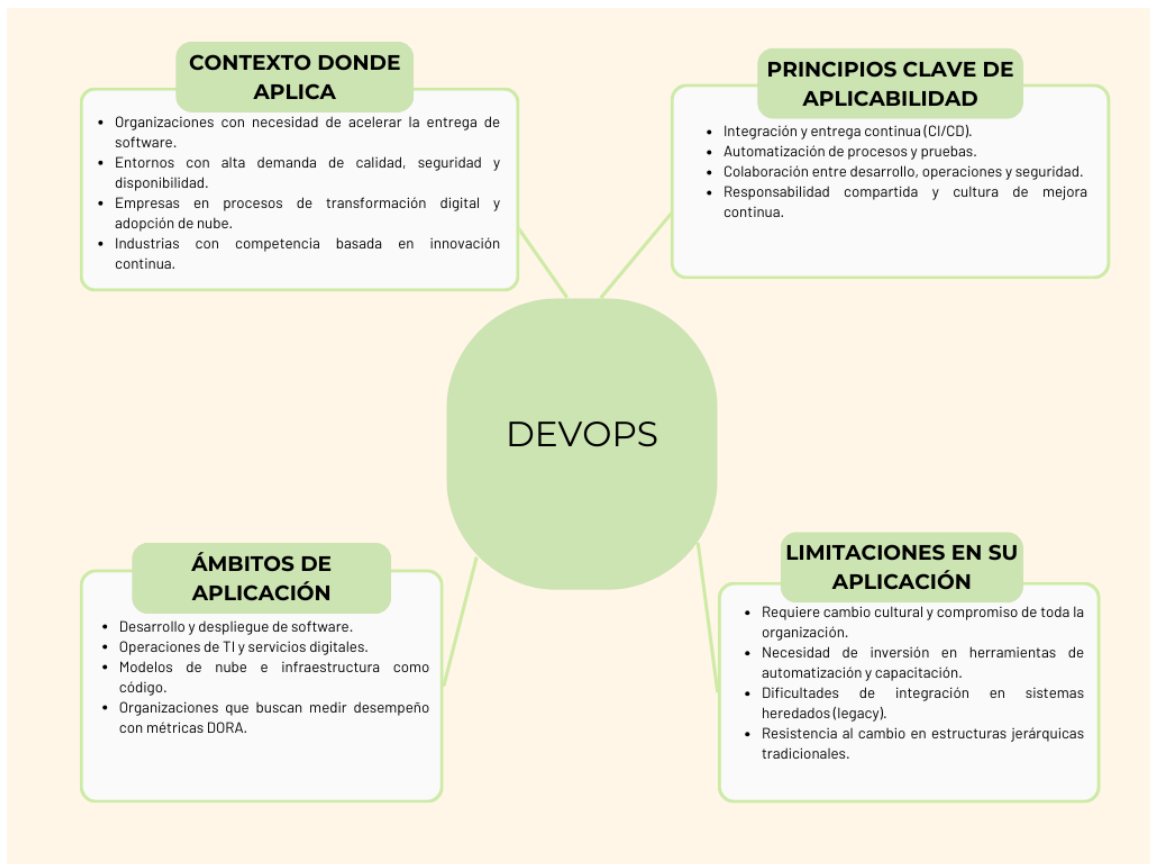
1. Reunión de Planificación (Sprint Planning Meeting): En esta reunión se definen las tareas que se van a realizar y los objetivos. Durante el desarrollo del Sprint, el equipo puede realizar consultas, autogestión del equipo, se puede realizar una nueva planificación. Si el desarrollo del Sprint no se puede realizar, el equipo no puede comprometerse a cumplir todo el Backlog. Se puede eliminar algún ítem para el Product Owner.
2. Reunión Diaria (Scrum Daily Meeting): En la reunión diaria, los equipos comparten la información relacionada con el desarrollo, dando un estatus de los avances e impedimentos que están teniendo. La reunión tiene un límite de tiempo de 15 minutos.
3. Reunión de revisión del sprint (Sprint Review Meeting): Esta reunión ayuda al equipo a tomar

decisiones para definir rutas para lograr cada una de las metas propuestas. En esta reunión es beneficiosa para todo el equipo, ya que se comprueba el cumplimiento de las expectativas y si se ha entendido la solicitud del cliente, produciendo una satisfacción en conjunto y validar el funcionamiento del entregable.

## 2.6.2 APLICABILIDAD DE DEVOPS

La aplicabilidad de DevOps radica en su capacidad para transformar el desarrollo, la entrega y la operación de software dentro de las organizaciones al integrar equipos que anteriormente estaban aislados, con el fin de lograr un flujo continuo de valor. Se utiliza en situaciones donde la velocidad, la calidad y la flexibilidad son importantes, como en proyectos de transformación digital, adopción de la nube, despliegue de servicios en línea y operaciones de TI que necesitan estar disponibles en todo momento.

**Figura 11** *Aplicabilidad de DevOps: contexto, principios, ámbitos y limitaciones*



Fuente: Adaptado de The DevOps Handbook (Kim et al., 2016) y literatura sobre prácticas DevOps y CI/CD.

### 2.6.3 APLICABILIDAD DE ITIL V4

ITIL V4 propone un marco flexible de gestión de servicios que incorpora la filosofía DevOps. De hecho, ITIL V4 se define como “una actualización del marco de mejores prácticas para la gestión de servicios de TI, que se centra en la integración de la gestión de servicios de TI con la gestión empresarial y ha sido influenciada por las metodologías ágiles y DevOps”. En otras palabras, ITIL V4 promueve un enfoque centrado en el cliente y la adaptabilidad, integrando de manera explícita principios ágiles y DevOps para hacer la entrega de servicios más eficiente y ágil.

Por ejemplo, Hiles y Barnes (2019) destacan que “ITIL V4 representa una mejora significativa con respecto a sus versiones anteriores, alineándose más estrechamente con las prácticas ágiles y DevOps”. Este enfoque modernizado permite a las organizaciones ser más ágiles y adaptativas en un entorno empresarial dinámico”.

Es importante mencionar que entre los principios guía de ITIL V4 se incluye expresamente “optimizar y automatizar”, reflejando el énfasis de DevOps en la automatización de pipelines y la eliminación de tareas manuales.

#### 2.6.3.1 BENEFICIOS DE ITIL V4 EN ENTORNOS DEVOPS

La aplicación de ITIL V4 en proyectos de desarrollo de software ofrece varios beneficios clave. En primer lugar, la estandarización de procesos operativos mediante ITIL V4 optimiza la utilización de recursos y agiliza la entrega de servicios. Como resumen, Rodríguez (2023):

La implementación de ITIL V4 permite optimizar los procesos operativos, al estandarizar flujos de trabajo y reducir redundancias. Mediante la automatización de tareas repetitivas como la generación de tickets, las organizaciones pueden minimizar los errores manuales y acelerar los tiempos de respuesta, incrementando la productividad de los equipos técnicos.

Desde la experiencia práctica, la adopción de ITIL V4 ha mostrado resultados tangibles. En un caso de estudio peruano se comprobó que un sistema de gestión de incidencias basado en ITIL V4 “ha permitido la sistematización efectiva de los procesos de resolución de problemas tecnológicos; se ha mejorado significativamente la efectividad en la identificación, clasificación y solución de incidentes, logrando una notable reducción en los tiempos de respuesta”. Esto evidencia que ITIL V4, correctamente implementado, reduce retrasos y refuerza el control operativo. Asimismo, la literatura internacional concluye que, “si se usa adecuadamente, el marco ITIL es importante para la gestión de servicios TIC, ya que proporciona resultados probados para el negocio, como reducción en el tiempo de ejecución de los procesos fortalecimiento del control

y monitoreo y una elevación notoria de la satisfacción de los clientes”.

#### 2.6.3.2 DESAFÍOS Y LIMITACIONES DE ITIL V4

La implementación de ITIL V4 también enfrenta obstáculos. Un reto frecuente es la resistencia cultural al cambio. Muchos profesionales están acostumbrados a flujos ágiles y pueden percibir ITIL como burocracia adicional. En palabras de Inoñan et al., en la práctica se ha observado que “hubo cierta reticencia inicial a invertir tiempo y recursos en lo que algunos consideraban una transformación innecesaria o compleja”. De hecho, autores como White (2018) advierten que “la resistencia al cambio y la falta de comprensión total de los principios de ITIL son obstáculos comunes”. Este choque cultural debe gestionarse con liderazgo y capacitación continua para que los equipos DevOps vean el valor de ITIL.

Otro desafío es integrar procesos formales de ITIL con ciclos rápidos de DevOps. La gestión de cambios tradicional de ITIL puede percibirse lenta frente a despliegues continuos. Como se señala en la práctica, “un desafío bastante común es integrar DevOps con prácticas existentes de ITIL/gestión de cambios en entornos corporativos muy estructurados”. En un entorno de desarrollo ágil, los procesos de aprobación y registro de cambios deben adaptarse (por ejemplo, automatizando los flujos de cambios con herramientas CI/CD) para no convertirse en cuello de botella. Adicionalmente, ITIL V4 suele requerir inversión inicial en herramientas ITSM (ServiceNow, Jira Service Management, etc.) y formación de personal; esto puede impactar los recursos de la consultora.

#### 2.6.3.3 PRÁCTICAS DE ITIL 4 Y SU APLICABILIDAD EN ENTORNOS DEVOPS

Según Sambrano Echave (2023), “ITIL v4 integra prácticas modernas que promueven la colaboración, la adaptabilidad y la generación de valor, lo cual resulta esencial en entornos dinámicos donde las organizaciones deben responder rápidamente a los cambios” (p. 45). Esta cita literal refleja la pertinencia de su adopción en contextos como el de NGS, donde la agilidad y la eficiencia en la entrega continua son factores críticos.

**Tabla 10** *Prácticas de ITIL 4 y su aplicabilidad en entornos DevOps*

<b>Categoría</b>	<b>Prácticas ITIL V4</b>	<b>Aplicabilidad a DevOps/NGS</b>
<b>Gestión general</b>	Gestión de la estrategia, arquitectura, mejora continua, gestión de riesgos, gestión de seguridad, gestión del conocimiento.	Mejora la planificación estratégica de TI y refuerza la seguridad en procesos de entrega continua.
<b>Gestión de servicios</b>	Gestión de incidentes, problemas, cambios, activos, configuraciones, solicitudes de servicio, niveles de servicio.	Fundamental para estandarizar flujos de soporte en NGS y asegurar despliegues estables en DevOps.
<b>Prácticas técnicas</b>	Desarrollo de software, gestión de despliegues, gestión de infraestructura, monitoreo y validación.	Facilita la integración con pipelines CI/CD, reduciendo errores y mejorando la automatización.

Fuente: Elaboración propia con base en Axelos (2019) e Inoñan (2023).

En este sentido, ITIL 4 no solo ofrece un marco de referencia amplio, sino que permite seleccionar prácticas específicas que aportan valor directo a la tesis. Para NGS destacan especialmente:

- **Gestión de Cambios:** asegura que las modificaciones en código o infraestructura se realicen de forma controlada.
- **Gestión de Incidentes y Problemas:** reduce el impacto de fallas en entornos de producción, alineado a la entrega continua.
- **Desarrollo de software y gestión de despliegues:** directamente relacionadas con la automatización y cultura DevOps.

Como señalan Valero (2025), “la alineación entre las prácticas de ITIL y la automatización que promueve DevOps genera un marco robusto para garantizar calidad, seguridad y continuidad en los servicios de TI” (p. 88).

## 2.7 INSTRUMENTOS UTILIZADOS

### 2.7.1 HERRAMIENTAS PROPIAS DE SCRUM

Lista de requisitos priorizados (Product Backlog): El Product Backlog es una lista priorizada que refleja lo que el cliente espera del proyecto, creada con apoyo del Scrum Máster y del equipo, donde se detalla el valor y el esfuerzo de cada requisito; no necesita estar completa desde el inicio, sino que se va ajustando con el avance, lo que permite enfocarse primero en lo más importante, adaptarse a cambios, evitar perder tiempo en tareas poco útiles y asegurar que cada entrega esté realmente terminada y lista para generar valor inmediato.

Lista de tareas de la iteración (Sprint Backlog): es la lista de tareas que el equipo prepara para cumplir los requisitos seleccionados en cada iteración, comprometiéndose a entregar un incremento de producto listo para el cliente; incluye las tareas, el esfuerzo pendiente y la autoasignación de cada miembro, lo que permite identificar bloqueos y tomar decisiones, mientras que el avance se controla mediante la gráfica burndown, que muestra la velocidad de progreso y las horas o tareas restantes hasta completar la iteración.

Gráficos de Trabajo pendiente (burndown): Es una herramienta visual que muestra, a lo largo del tiempo, cómo el equipo avanza en la finalización de los requisitos durante una iteración. Su función principal es evidenciar si el equipo podrá terminar el trabajo dentro del plazo estimado, ya que permite ver de un vistazo: cuánto trabajo ya se ha completado, cuánto queda pendiente, la velocidad de progreso y la fecha en la que se espera finalizar.

En esencia, este gráfico facilita el seguimiento del ritmo del equipo y la toma de decisiones oportunas para ajustar prioridades o resolver bloqueos, asegurando que la entrega de valor al cliente se cumpla en los tiempos comprometidos.

**Figura 12** *Gráfica Burndown*



Fuente (Pérez Pérez, s.f., p. 30)

- En el eje Y, cantidad de trabajo pendiente de terminar - En el eje X, tiempo (iteraciones). Este tipo de gráfico permite realizar diversas simulaciones: ver cómo se aplazan las fechas de entrega, si se le añaden requisitos, ver cómo se avanzan, si se les quitan requisitos o se añade otro equipo, etc.

### 2.7.2 HERRAMIENTAS Y PRÁCTICAS ASOCIADAS A DEVOPS

DORA es un marco de métricas creado por DevOps Research and Assessment, una organización de investigación que estudió durante años las prácticas de alto rendimiento en equipos de software. Su mayor aporte son las 4 métricas clave de desempeño DevOps, que permiten medir la efectividad de los procesos de entrega continua.

### 2.7.3 HERRAMIENTAS Y PRÁCTICAS DE ITIL V4

#### 2.7.3.1 PRÁCTICAS DE ITIL COMO INSTRUMENTOS DE GESTIÓN

Las 34 prácticas de ITIL v4 se constituyen como los principales instrumentos para aplicar en las organizaciones. Estas prácticas abarcan gestión general, gestión de servicios y gestión técnica.

**Tabla 11** *Clasificación de las prácticas de ITIL v4*

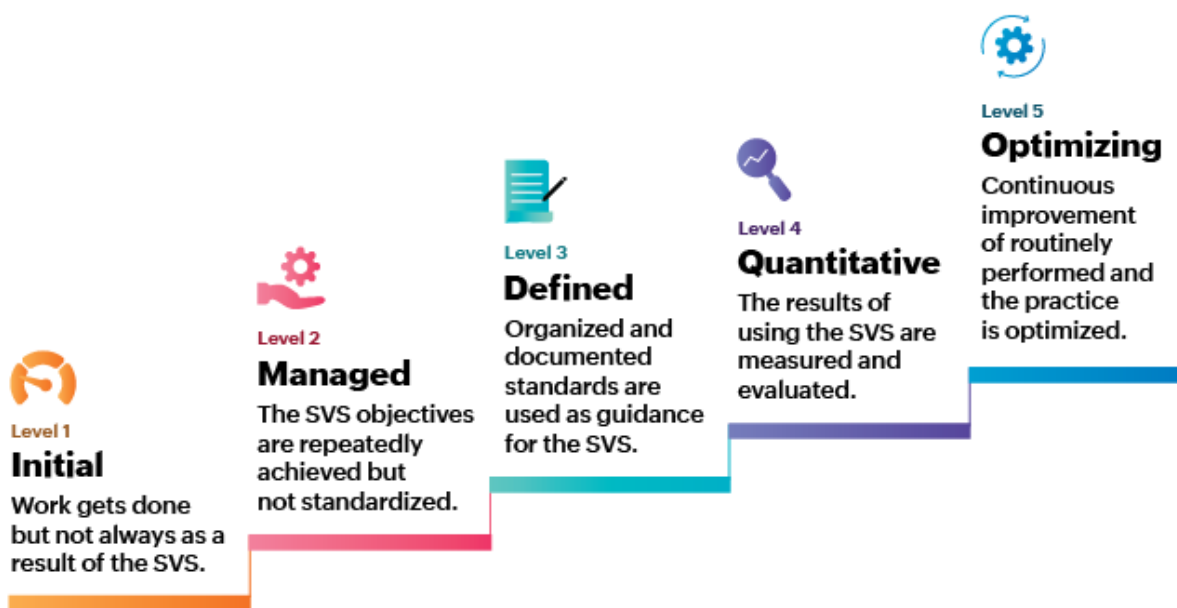
Categoría	Prácticas principales
<b>Gestión general</b>	<ul style="list-style-type: none"> <li>• Mejora continua</li> <li>• Gestión del cambio organizacional</li> <li>• Gestión de riesgos</li> <li>• Medición e informes</li> <li>• Gestión de portafolio</li> </ul>
<b>Gestión de servicios</b>	<ul style="list-style-type: none"> <li>• Mesa de servicio</li> <li>• Gestión de incidentes</li> <li>• Gestión de problemas</li> <li>• Gestión de solicitudes de servicio</li> <li>• Gestión de niveles de servicio (SLAs)</li> </ul>
<b>Gestión técnica</b>	<ul style="list-style-type: none"> <li>• Gestión de despliegues</li> <li>• Gestión de infraestructura y plataformas</li> <li>• Gestión de software</li> <li>• Gestión de la nube</li> <li>• Gestión de seguridad de la información.</li> </ul>

Fuente: Elaboración propia a partir de Axelos, 2019; Sambrano, 2023.

### 2.7.3.2 MATRIZ DE MADUREZ DE PRÁCTICAS ITIL V4

La Matriz de Madurez de Prácticas ITIL V4, ayuda a medir el grado de adopción y estandarización las prácticas en la Consultora Nueva Generación de Software (NGS). La matriz se basa en el ITIL Maturity Model (ITIL MM) de PeopleCert, una herramienta desarrollada para evaluar de forma objetiva la capacidad y madurez de prácticas individuales o del SVS en su totalidad.

**Figura 13** *ITIL Maturity Model*



Fuente: Adaptado de PeopleCert (2025).

En esta investigación, se adapta el modelo para evaluar prácticas clave de gestión de cambios, incidentes, despliegues y mejora continua. Estas son esenciales para integrar metodologías ágiles y prácticas DevOps en ITIL. En esta investigación, la adaptación del modelo se centra en evaluar prácticas esenciales de gestión de cambios, incidentes, despliegues y mejora continua, fundamentales para la integración de metodologías ágiles y prácticas DevOps en ITIL V4. La matriz elaborada se utilizó como herramienta de diagnóstico para demostrar el nivel actual de madurez en NGS y planificar las acciones para movernos hacia un modelo de integración más eficiente y en línea con los estándares internacionales de gestión de servicios.

## 2.8 CONCEPTUALIZACIÓN

La presente sección de conceptualización reúne y define de manera sintética los principales conceptos desarrollados en el Capítulo II, con el propósito de unificar criterios de interpretación y asegurar un uso consistente de la terminología a lo largo de la investigación. Las definiciones se presentan desde un enfoque operativo, alineado con el contexto del estudio, sin sustituir el desarrollo teórico previamente expuesto.

**DevOps:** Es un enfoque cultural y técnico que integra el desarrollo de software y las operaciones de TI, promoviendo la colaboración, la automatización y la mejora continua para acelerar la entrega de valor con calidad y confiabilidad (*Kim et al., 2016*).

**Metodologías ágiles:** Corresponden a enfoques iterativos e incrementales que priorizan la adaptación al cambio, la colaboración entre equipos y la entrega continua de valor al cliente (*Beck et al., 2001*).

**Scrum:** Es un marco de trabajo ágil que permite gestionar proyectos complejos mediante iteraciones cortas, roles definidos y eventos que facilitan la inspección y adaptación continua (*Schwaber & Sutherland, 2020*).

**Eventos de Scrum:** Establecen una cadencia regular que facilita la inspección y adaptación del trabajo, reduciendo la necesidad de reuniones adicionales (*Schwaber & Sutherland, 2020*).

**Sprint:** Es un periodo de tiempo fijo durante el cual se desarrolla un incremento de producto potencialmente entregable (*Schwaber & Sutherland, 2020*).

**Scrum Team:** Es una unidad autoorganizada y multifuncional compuesta por Product Owner, Scrum Master y Developers, responsable de generar valor en cada Sprint (*Schwaber & Sutherland, 2020*).

**Product Owner:** Es responsable de maximizar el valor del producto mediante la gestión del Product Backlog (*Schwaber & Sutherland, 2020*).

**Development Team:** Los Developers son responsables de crear los incrementos de producto y cumplir los objetivos definidos para el Sprint (*Schwaber & Sutherland, 2020*).

**Scrum Máster:** El Scrum Master facilita la correcta aplicación de Scrum, promoviendo la mejora continua y la eliminación de impedimentos (*Schwaber & Sutherland, 2020*).

**Sprint Planning:** Evento en el cual el Scrum Team define el objetivo del Sprint y el trabajo necesario para alcanzarlo (*Schwaber & Sutherland, 2020*).

**Daily Scrum:** Reunión diaria breve donde el equipo inspecciona el progreso y ajusta su plan de trabajo (*Schwaber & Sutherland, 2020*).

**Sprint Review:** Evento realizado al final del Sprint para inspeccionar el incremento y adaptar el Product Backlog (*Schwaber & Sutherland, 2020*).

**Sprint Retrospective:** Evento enfocado en identificar oportunidades de mejora en personas, procesos y herramientas (*Schwaber & Sutherland, 2020*).

**Kanban:** Es una herramienta ágil de gestión de proyectos diseñada para ayudar a visualizar el trabajo, limitar el trabajo en curso y maximizar la eficiencia (o el flujo). Rehkopf, M. (2025)

Acevedo et al. establecen que Kanban es:

Una técnica de gestión de producción basada en un sistema pull (halar) que se fundamenta en la autogestión de los procesos, eliminando la programación centralizada. Se produce y transporta lo que se demanda en los procesos consumidores, manteniendo en rotación solo aquellas cantidades que garantizan la continuidad del consumo. Cuando se interrumpe el consumo, se detiene la producción. Es una herramienta para conseguir la producción justo a tiempo.

**Integración Continua (CI):** La integración continua es la práctica de integrar frecuentemente cambios de código en un repositorio compartido con validación automática (*Fowler, 2006*).

**Entrega Continua (CD):** La entrega continua mantiene el software en un estado desplegable mediante procesos automatizados (*Humble, 2010*).

**Infraestructura como Código (IaC):** Es la gestión de la infraestructura tecnológica (aprovisionamiento de máquinas virtuales, servidores, redes, instalación y configuración de paquetes de software) a través de código (scripts) y uso de prácticas tradicionales de la Ingeniería de Software, como la gestión de calidad de código, pruebas, control de versión (*Guckenheimer, 2017*), (*Sharma, Fragkoulis, & Spinellis, 2016*).

**Mejora Continua:** Es un enfoque sistemático que busca optimizar los procesos y servicios de manera constante, mediante la evaluación y adaptación continua para satisfacer las cambiantes

necesidades del negocio y del cliente (Axelos, 2020).

**Transformación digital:** Se ha consolidado como un fenómeno trascendental en la era contemporánea, impulsado por la proliferación de tecnologías digitales, automatización, inteligencia artificial, internet de las cosas, blockchain, los gemelos digitales y la modesta pero creciente integración en todos los aspectos de la sociedad y la economía (CEPAL, 2022).

**Calidad de software:** La calidad de software es el grado en que un producto satisface requisitos funcionales y no funcionales (ISO/IEC, 2011).

**DevSecOps:** Según AWS (Amazon Web Services). (s. f.):

DevSecOps es la práctica de integrar las pruebas de seguridad en cada etapa del proceso de desarrollo de software. Incluye herramientas y procesos que fomentan la colaboración entre los desarrolladores, los especialistas en seguridad y los equipos de operaciones para crear un software que sea eficiente y seguro.

**Docker:** Es una plataforma que permite empaquetar aplicaciones en contenedores ligeros para facilitar su despliegue y portabilidad (Merkel, 2014).

**Kubernetes:** Es un sistema de orquestación de contenedores que automatiza despliegue, escalado y operación de aplicaciones (*Burns et al., 2016*).

**Eficiencia operativa:** Capacidad de una organización para ejecutar sus procesos de la manera más efectiva y económica posible. En términos simples, “la eficiencia operativa trata en que nuestra competencia no realice las tareas de una mejor forma que nosotros, capaz de usar de manera inmejorable los recursos” (Leal & Quispe, 2018, p. 39).

**Cultura colaborativa:** Clima organizacional que valora la cooperación, la comunicación abierta y el trabajo conjunto entre diferentes áreas o equipos. Por ejemplo, en un entorno DevOps “se fomenta la colaboración para agilizar la respuesta a cambios, requerimientos y actualizaciones” (Azad & Hyrynsalmi, 2023), reflejando la importancia de compartir información y objetivos comunes.

**Herramientas CI/CD:** Plataformas que implementan los procesos de integración y entrega continua, automatizando la construcción, prueba y despliegue del software. Aguilar Riera señala que “se ha seleccionado GitLab CI/CD para la automatización del ciclo de vida del software. GitLab CI/CD proporciona un marco” (Aguilar Riera, 2025), lo que ilustra cómo estas

herramientas agilizan el ciclo de desarrollo y entrega.

**Rol del ingeniero DevOps:** El ingeniero DevOps es un profesional que combina habilidades de desarrollo y operaciones para facilitar la integración, entrega y despliegue continuos de software, mediante la implementación y automatización de herramientas, colaboración entre equipos y mejora continua de procesos tecnológicos. Este rol se centra en optimizar la eficiencia del flujo de trabajo, la confiabilidad del software y la respuesta ante cambios o incidentes, alineando prácticas de desarrollo con operaciones de TI (Atlassian, s. f.).

**NGS:** Nueva Generación de Software.

**ITSM:** (Information Technology Service Management), o Gestión de Servicios de Tecnologías de la Información, es un enfoque estratégico y estructurado para diseñar, entregar, gestionar y mejorar los servicios de TI en una organización.

Según (Marrone & Kolbe, 2011), "ITSM es un enfoque orientado a procesos para gestionar los servicios de tecnologías de la información, que busca mejorar la alineación de TI con el negocio a través de la implementación de prácticas estructuradas".

**SVS:** El SVS describe cómo los componentes organizacionales trabajan juntos para facilitar la creación de valor (AXELOS, 2019).

**Prácticas de ITIL 4:** ITIL 4 define prácticas organizacionales orientadas a la gestión integral de servicios (AXELOS, 2019).

**Gestión de problemas:** La gestión de problemas busca identificar y eliminar causas raíz de incidentes (AXELOS, 2019).

**Gestión de cambios:** La gestión de cambios asegura que las modificaciones se realicen de manera controlada (AXELOS, 2019).

**Valor del servicio:** El valor del servicio se define como el equilibrio entre beneficios, costos y riesgos (AXELOS, 2019).

## 2.9 MARCO LEGAL Y NORMATIVO

### 2.9.1 NORMATIVA INTERNACIONAL

#### 2.9.1.1 REGULACIONES RELACIONADAS CON SCRUM

En el caso de las metodologías ágiles como Scrum, a nivel internacional, no existen marcos legales

específicos que estén regulados por un tratado o ley universal. Existen normas y estándares internacionales relacionados como son:

- **ISO 21502:2020** (*Project, programme and portfolio management*) establece directrices aplicables a proyectos gestionados con metodologías ágiles.
- **ISO/IEC 12207:2017** regula procesos de ciclo de vida de software, donde Scrum y Kanban pueden adaptarse como prácticas.
- **ISO/IEC/IEEE 26515:2018** (User documentation in agile environments) guía sobre documentación en proyectos ágiles.
- **PMI (Project Management Institute)** en la *PMBOK Guide* y en el *Agile Practice Guide* (2017, junto con Agile Alliance) estándares globales de referencia.

#### 2.9.1.2 REGULACIONES APLICABLES A DEVOPS

- **ISO/IEC 27001:2022** (seguridad de la información).
- **ISO/IEC 25010:2011** (calidad del software).
- **NIST DevSecOps Framework** (EE. UU.) como guía para integrar seguridad en el ciclo de vida del software.

#### 2.9.1.3 REGULACIONES VINCULADAS A ITIL V4

- ITIL se enmarca en estándares de gestión de servicios como la **ISO/IEC 20000-1:2018**, norma certificable para sistemas de gestión de servicios de TI (SGSTI).

### 2.9.2 NORMATIVA NACIONAL

En Honduras, no existe un marco legal específico o normativo nacional que regula o exija las implementaciones de las metodologías ágiles, ITIL 4, prácticas DevOps y COBIT 2019.

#### 2.9.2.1 MARCO NORMATIVO EN SCRUM

- No existe normativa que regule las metodologías ágiles en Honduras; se está utilizando en diferentes empresas; ampliamente adoptado y enseñado.

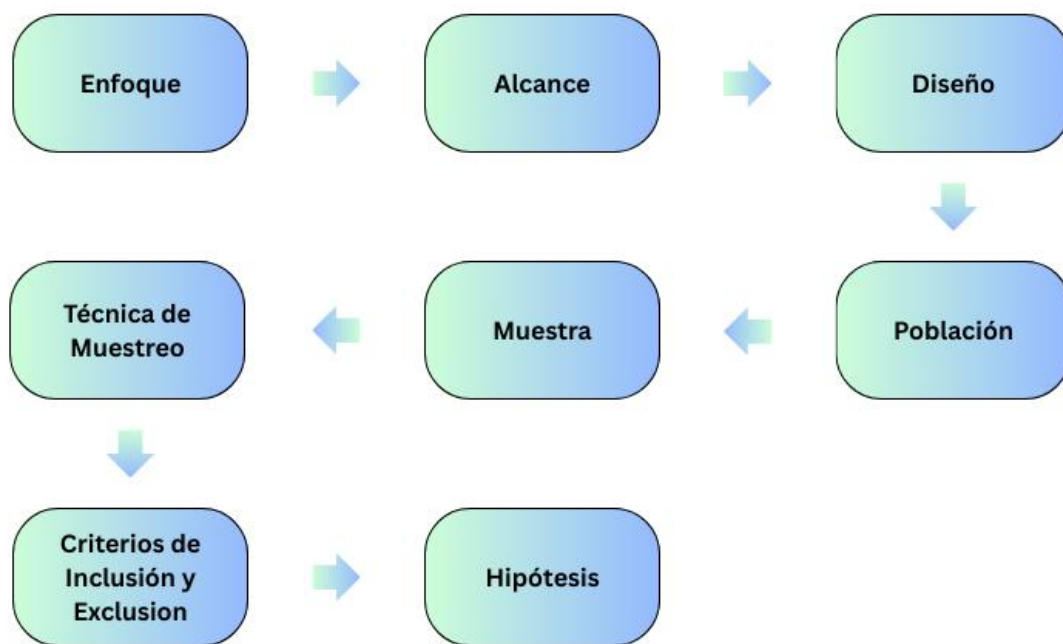
#### 2.9.2.2 MARCO NORMATIVO EN ITIL V4

- No existe una normativa estatal que imponga o defina ITIL 4, se aplica a la práctica de marco flexible.

## CAPÍTULO III - METODOLOGÍA DE LA INVESTIGACIÓN

La metodología de la investigación es una guía general que dirige el enfoque de problemas científicos a través de métodos estructurados. Bunge (2013) la define como un procedimiento que comprende la formulación de hipótesis, el planteamiento de problemas, la creación de modelos teóricos y la integración de conclusiones, lo cual hace que sea un marco fundamental para crear conocimiento (p. 24).

**Figura 14** Orden lógico para la metodología de investigación



Nota: Elaboración Propia.

### 3.1 ENFOQUE

Creswell (1998) sostiene que la investigación cualitativa es un proceso de indagación interpretativo, apoyado en diversas tradiciones metodológicas, como la teoría fundamentada en los datos, la etnografía, el estudio de casos, la biografía y la fenomenología. Su propósito es analizar un problema social o humano. La persona que investiga crea una imagen compleja y global, examina palabras, ofrece perspectivas minuciosas de los informantes y lleva a cabo el estudio en un contexto natural. Sandín (2003) sostiene que busca entender a fondo los fenómenos sociales y educativos, cambiar las prácticas y los contextos socio educativos, tomar decisiones y descubrir y

desarrollar un conjunto de conocimientos organizado. Según Denzin & Lincoln (2018)

La investigación cualitativa es una actividad que sitúa al observador en el mundo. La investigación cualitativa consiste en un conjunto de prácticas interpretativas y materiales que hacen visible el mundo. Estas prácticas transforman el mundo en una serie de representaciones, como notas de campo, entrevistas, conversaciones, fotografías, grabaciones y notas para el yo (investigador).

La investigación cuantitativa se caracteriza por el uso de mediciones precisas y análisis estadísticos para explicar fenómenos mediante relaciones entre variables, apoyándose en una lógica deductiva y en el uso de instrumentos estandarizados que permiten la replicación y comparación de resultados (Forni & De Grande, 2020; Sánchez et al., 2021).

El presente estudio adopta un enfoque mixto con predominio cualitativo, ya que combina técnicas cualitativas y cuantitativas con el propósito de comprender, diagnosticar y proponer mejoras en los procesos, prácticas y capacidades de la consultora Nueva Generación de Software (NGS) en relación con la adopción de prácticas DevOps y el flujo de integración y entrega continua (CI/CD). El componente cualitativo constituye el eje central de la investigación, mientras que el componente cuantitativo se utiliza de forma complementaria y descriptiva para apoyar el análisis y la priorización de brechas.

**Tabla 12** *Definición del enfoque por cada objetivo específico*

Objetivo	Enfoque
Analizar qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps.	Mixto con predominio cualitativo
Identificar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI.	
Evaluar la arquitectura tecnológica más adecuada para soportar la implementación de un flujo de CI/CD en NGS, garantizando la escalabilidad, la seguridad y la trazabilidad.	
Proponer un conjunto de indicadores clave de desempeño (KPIs) que permitan evaluar de manera objetiva el impacto de la adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software y satisfacción del cliente.	

Nota: Elaboración propia basándonos en los objetivos específicos definidos en el capítulo I.

### **3.2 ALCANCE DE LA METODOLOGÍA**

El tipo o alcance de la investigación es subsecuente a la definición del enfoque investigativo, dado que los mismos sirven como complemento de los resultados que se obtendrán y el sentido que tomará la investigación; precisamente, el término tipos alude a la caracterización que tendrá el conocimiento a obtener, mientras que el nivel hace eco de cuál será el alcance que tendrá la investigación, es decir, los límites, generando espacios para que se siga expandiendo el conocimiento a partir de nuevas investigaciones (Hernández et al., 2010).

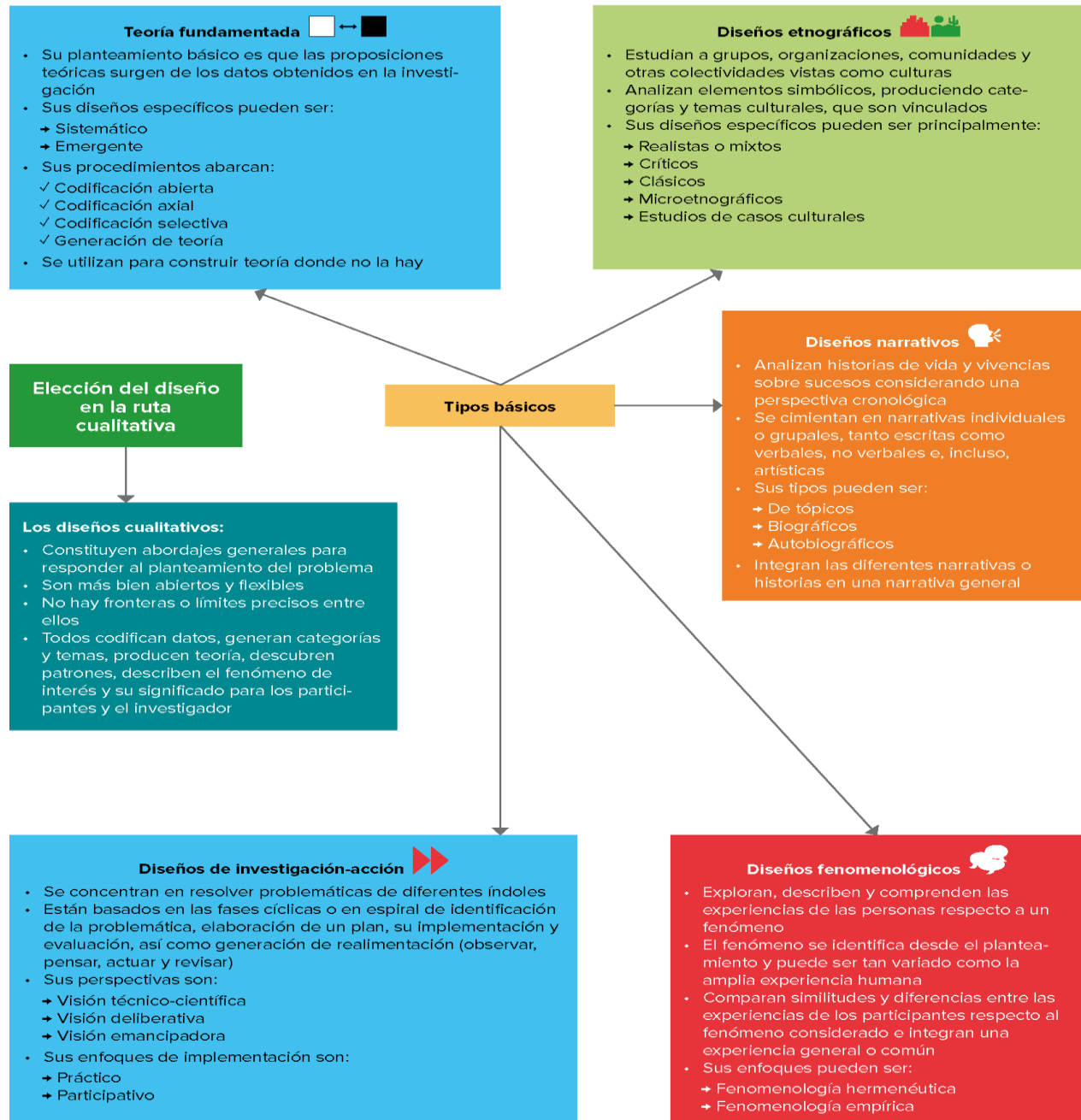
Nuestra investigación tiene un alcance exploratorio-descriptivo, puesto que su objetivo es determinar y entender las diferencias en los procesos, herramientas y competencias de NGS con respecto a la implementación de prácticas DevOps y metodologías ágiles bajo el marco ITIL V4. En su aspecto exploratorio, la investigación pretende ofrecer un espacio de análisis inicial que facilite la detección de oportunidades para estandarizar y optimizar la entrega continua de software. Al mismo tiempo, su naturaleza descriptiva se manifiesta en la sistematización de los procedimientos actuales de NGS con el fin de identificar sus puntos fuertes y débiles y establecer los cimientos de una propuesta metodológica que guíe a la organización hacia un modelo más eficaz, rápido y acorde con las normas internacionales.

### **3.3 DISEÑO**

El diseño se refiere al abordaje general que habrás de utilizar en el proceso de investigación. Miller y Crabtree (1992) lo denominan enfoque, Álvarez-Gayou (2003) marco interpretativo y Denzin y Lincoln (2005) estrategia de indagación.

El diseño metodológico de esta investigación se basa en la teoría fundamental y la investigación-acción. La teoría fundamental posibilita el reconocimiento y la clasificación de las brechas en los procesos, herramientas y competencias de NGS, lo que permite esclarecer su efecto sobre la implementación de DevOps e ITIL V4. La investigación-acción, por su parte, incorpora un aspecto práctico, porque la propuesta tiene como objetivo no solamente hacer un diagnóstico, sino también orientar una planificación de implementación gradual que trate las necesidades reales de la organización.

**Figura 15 Elección del Diseño en un Enfoque Cualitativo**



Fuente: Adaptado desde Hernández Sampieri, R., Mendoza Torres, C. P.(2023). Metodología de la Investigación. McGraw-Hill Interamericana.

### 3.4 POBLACIÓN

Varios autores han abordado la definición de población, enfatizando su importancia para el establecimiento de límites y la comprensión del objeto que se está analizando. Hernández et al. (2014) la caracterizan como el conjunto de individuos o eventos que tienen características

específicas en común y que son objeto de investigación. Esta definición pone de relieve el carácter integral de la población, que abarca a todos los elementos o individuos que cumplen con ciertos criterios.

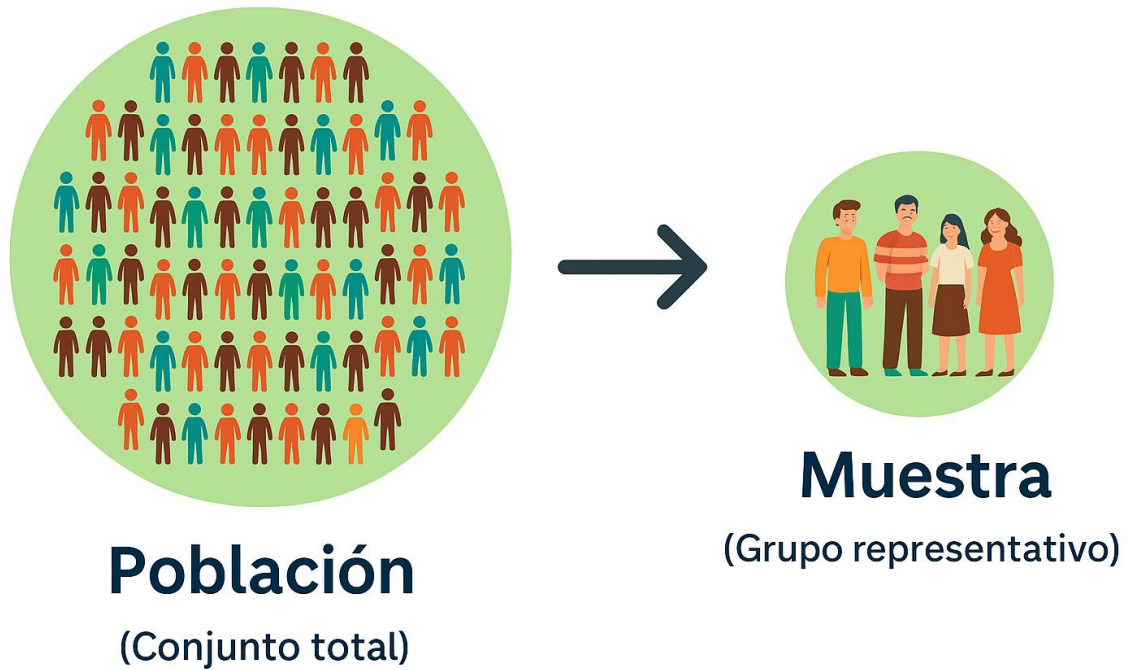
Según lo señalan Rebollo y Ábalos (2022), la selección de la población en estos estudios se centra en personas, acontecimientos o lugares que ofrecen una vasta cantidad de datos y permiten obtener un punto de vista detallado del fenómeno a investigar. Este muestreo, que se denomina también muestreo dirigido o intencionado, tiene como objetivo determinar quiénes son los participantes con experiencias, conocimientos o perspectivas concretas que sean relevantes para el tema de investigación (Creswell, 2013).

La población de este estudio está compuesta por los 60 trabajadores de la Consultora Nueva Generación de Software (NGS), que intervienen en las etapas de desarrollo, pruebas, gestión y aseguramiento de calidad. Esta población incluye a los equipos técnicos y de administración que participan directa o indirectamente en la adopción de las prácticas DevOps e ITIL V4. Asimismo, se incorporan 6 profesionales pertenecientes a una organización cliente de NGS, quienes mantienen una relación activa con la gestión de los proyectos desarrollados por la consultora y aportan una perspectiva externa sobre la aplicación de dichas prácticas. También cuenta con diferentes procesos, pero para este estudio solo se considerará el proceso de gestión integral del ciclo de vida del software, ya que forma parte del alcance de nuestro estudio.

### **3.5 MUESTRA**

Dentro del ámbito de la investigación, una muestra se define como un subconjunto seleccionado de individuos, eventos o entidades extraídos de una población total o universo, con el propósito de llevar a cabo un estudio y hacer inferencias sobre dicha población (Arias y Covinos, Diseño y metodología de la investigación, 2021). Este concepto es fundamental en la metodología de la investigación, ya que una adecuada selección de la muestra puede determinar la validez y fiabilidad de los resultados obtenidos. La muestra permite a los investigadores estudiar y analizar características específicas de un grupo representativo sin necesidad de examinar a cada miembro de la población, lo que, en muchas ocasiones, resultaría impráctico o incluso imposible debido a limitaciones de tiempo, recursos o acceso (Sekaran y Bougie, 2016). Es esencial, por tanto, que la muestra se seleccione con rigor metodológico para garantizar que sea un reflejo adecuado y representativo del grupo o población en su totalidad.

**Figura 16** *Relación entre población y muestra en investigación*



Nota: Elaboración Propia.

Para nuestro caso, la muestra está conformada por 26 colaboradores de la Consultora Nueva Generación de Software (NGS), elegidos de un total de 60 trabajadores repartidos en varias áreas y equipos esenciales de la empresa. Asimismo, se incluyen 2 representantes de una entidad cliente de NGS, con el propósito de incorporar una perspectiva externa sobre la gestión y ejecución de los proyectos. La razón de optar por esta muestra es que su tamaño es manejable y permite obtener una visión completa acerca de los procedimientos y prácticas actuales en desarrollo, aseguramiento de calidad, administración de proyectos, gerencia y recursos humanos. Para los procesos se considerará como parte de la muestra la gestión integral del ciclo de vida del software en NGS.

**Tabla 13** *Distribución de la población y muestra por área*

Áreas	Número de involucrados en la población	Número de involucrados en la muestra
Gerencia	1	1
Dirección	1	1
Recursos Humanos	2	0

Desarrolladores	38	19
QA	5	2
Gestión de proyectos	1	0
Otras Áreas	12	0
Clientes (externos)	6	3
Total	66	26

Nota: Elaboración propia.

**Tabla 14** *Resumen de la población y muestra de la investigación*

Dimensión	Población	Muestra
Personas	66	26
Procesos	1	1

Nota: Elaboración propia.

### 3.6 TÉCNICA DE MUESTREO

Otra implicancia de la correcta identificación de la población es la determinación de las técnicas estadísticas apropiadas para el análisis. Por ejemplo, cuando se trabaja con poblaciones finitas, algunas técnicas de muestreo y análisis pueden ser más adecuadas que cuando se trabaja con poblaciones infinitas (Morillas, 2007).

Por lo tanto, este estudio utilizará un muestreo no probabilístico intencionado, ya que el personal de NGS está compuesto por 60 miembros que trabajan en distintos campos y un proceso integral de gestión del ciclo de vida del software. De este total, 26 participantes se seleccionaron debido a su papel fundamental en las funciones vinculadas con el desarrollo, la garantía de calidad y la gestión de proyectos. Para ofrecer una visión estratégica, también se incluyen a los representantes de la dirección, gerencia y recursos humanos. Esta visión asegura que los datos recolectados provengan de actores que están directamente implicados en procesos críticos, permitiendo así la adquisición de información más relevante y minuciosa. De esta manera, se trata el carácter cualitativo de la investigación, donde lo que más importa no es la generalización estadística, sino comprender las oportunidades de mejora y las brechas en los procedimientos de NGS.

### 3.7 CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN

#### 3.7.1 PERSONAL

Los criterios de inclusión y exclusión se presentan en conjunto para mostrar de manera contrastada a los colaboradores que aportan información relevante para el estudio y aquellos cuya

participación no resultaría pertinente, ya que se trata de aspectos complementarios en la delimitación de la muestra.

**Tabla 15** *Criterios de inclusión y exclusión según la muestra para el personal*

<b>Criterios de Inclusión</b>	<b>Criterios de Exclusión</b>
Colaboradores que desempeñan roles directamente involucrados en el desarrollo de software, aseguramiento de calidad y gestión de servicios.	Colaboradores que no desempeñan roles directamente involucrados en el desarrollo de software, aseguramiento de calidad y gestión de servicios.
Participantes con conocimientos básicos de los principios DevOps y/o el uso de marcos ITIL.	Participantes sin conocimientos básicos de los principios DevOps y/o el uso de marcos ITIL.
Empleados con entendimiento amplio de los procesos internos y la cultura de NGS.	Empleados que, pese a trabajar en NGS, no participan en los procesos de desarrollo, entrega continua, despliegue ni en la gestión de servicios TI.
Individuos que expresen consentimiento y dispongan de tiempo dentro de su jornada laboral o no laboral para participar.	Colaboradores que no pueden comprometerse con las actividades del estudio.
Colaboradores externos que manifiesten disposición para participar en el estudio y aportar su punto de vista sobre la gestión de proyectos desarrollados por NGS.	Colaboradores externos que no manifiesten disposición para participar en el estudio y aportar su punto de vista sobre la gestión de proyectos desarrollados por NGS.

Nota: Elaboración propia.

### 3.7.2 PROCESOS

Para esta investigación se tomarán en cuenta únicamente los procesos vinculados al desarrollo y entrega de software en la Consultora Nueva Generación de Software (NGS).

**Tabla 16** *Criterios de inclusión y exclusión según la muestra para procesos*

<b>Criterios de Inclusión</b>	<b>Criterios de Exclusión</b>
Procesos directamente vinculados al ciclo de vida del software como desarrollo, pruebas, aseguramiento de calidad, despliegue y gestión de servicios.	Procesos no vinculados al ciclo de vida del software como desarrollo, pruebas, aseguramiento de calidad, despliegue y gestión de servicios.

Procesos transversales que impactan la entrega de valor al cliente, como la gestión de proyectos y la gestión de cambios.	Procesos que no impactan la entrega de valor al cliente, como la gestión de proyectos y la gestión de cambios.
Procesos con evidencia documentada y métricas disponibles (reportes de QA, indicadores de incidencias, SLA).	Procesos sin interacción con desarrollo, QA o gestión de servicios, cuya participación no aporta información relevante al objeto de estudio.

Nota: Elaboración propia.

### 3.8 HIPÓTESIS

La formulación de hipótesis constituye un componente fundamental del enfoque cuantitativo y del componente cuantitativo dentro de los estudios de enfoque mixto, ya que permite establecer relaciones claras y medibles entre variables con el propósito de orientar el análisis y la evaluación de los resultados. Según Forni y De Grande (2020), las hipótesis se formulan bajo una lógica deductiva y deben ser susceptibles de contrastación empírica mediante indicadores objetivos y sistemáticos. De igual forma, Bagur-Pons et al. (2021) señalan que las hipótesis deben definirse de manera operativa y medible para facilitar su validación, especialmente en investigaciones aplicadas orientadas a la evaluación de mejoras en desempeño, eficiencia y calidad. En este contexto, la incorporación de hipótesis resulta pertinente cuando se busca analizar el impacto potencial de una propuesta metodológica mediante métricas de desempeño, aun cuando el diseño general del estudio mantenga un predominio cualitativo.

En el presente estudio se incorpora una hipótesis de investigación como parte del componente cuantitativo del enfoque mixto adoptado, con el propósito de orientar el análisis sobre los posibles efectos de la propuesta metodológica planteada. Aunque la investigación mantiene un predominio cualitativo y un alcance descriptivo-propositivo, las hipótesis se formulan como supuestos orientadores, sustentados en el diagnóstico realizado, las brechas identificadas y las buenas prácticas de DevOps, metodologías ágiles e ITIL v4, con el fin de estimar el impacto esperado del modelo propuesto en el desempeño del flujo de integración y entrega continua (CI/CD), particularmente en términos de tiempos de respuesta y calidad del software, sin pretender su comprobación mediante pruebas estadísticas estrictas.

Bajo este enfoque, se plantea la siguiente hipótesis de investigación:

La implementación estructurada de prácticas DevOps, integradas con metodologías ágiles y alineadas con ITIL v4, permitiría reducir los tiempos de respuesta ante cambios, incidencias y despliegues, así como mejorar la calidad de los entregables de software mediante una mayor automatización, estandarización y control del proceso CI/CD.

La hipótesis planteada en este estudio se formula como una hipótesis orientadora, dado que la investigación tiene un alcance descriptivo-propositivo y no contempla la implementación operativa del modelo durante su desarrollo. En consecuencia, dicha hipótesis no se somete a pruebas estadísticas de comprobación, sino que sirve como referencia conceptual para orientar el análisis, la definición de indicadores y la estimación del impacto esperado de la propuesta. La validación empírica de la hipótesis se plantea como una línea futura de investigación, una vez que el modelo sea implementado en la organización.

### **3.9 OPERACIONALIZACIÓN DE LAS VARIABLES**

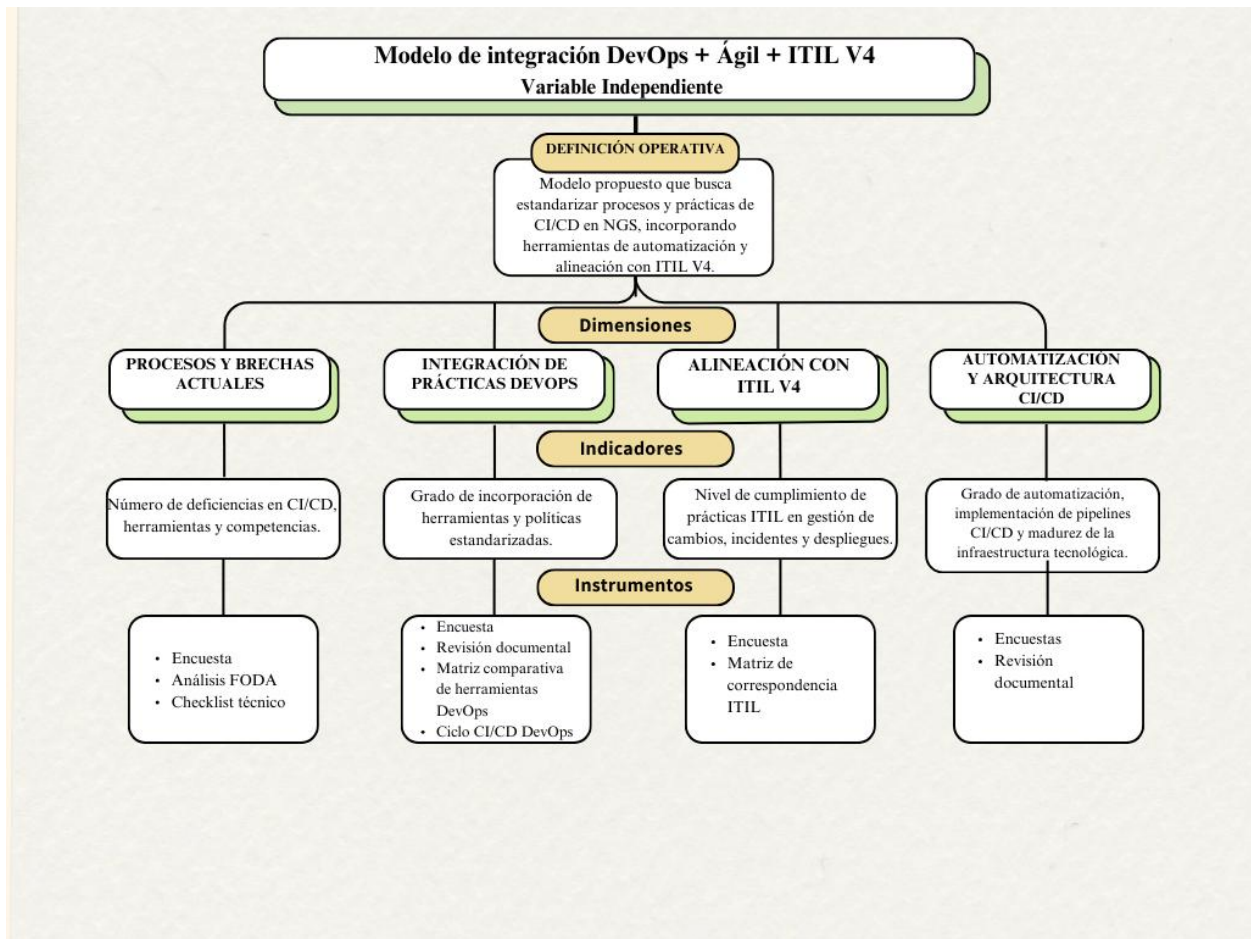
La operacionalización de variables es el proceso de convertir ideas abstractas en definiciones claras y observables. Esto establece cómo se medirán o identificarán en un análisis. Desde una perspectiva metodológica, esta etapa permite que la teoría y la realidad observable se vinculen a través de la formulación de indicadores, criterios de medición o dimensiones para cada variable fundamental (Coronel-Carvajal, 2023).

Según Coronel-Carvajal (2023):

Del correcto manejo de las variables depende la veracidad a los resultados de la investigación y la validez del instrumento para la recolección de datos. La operacionalización de variables consiste en un conjunto de técnicas y métodos que permiten medir la variable en una investigación. Es un proceso de separación y análisis de la variable en sus componentes que permiten medirla (Coronel-Carvajal, 2023, p. 1).

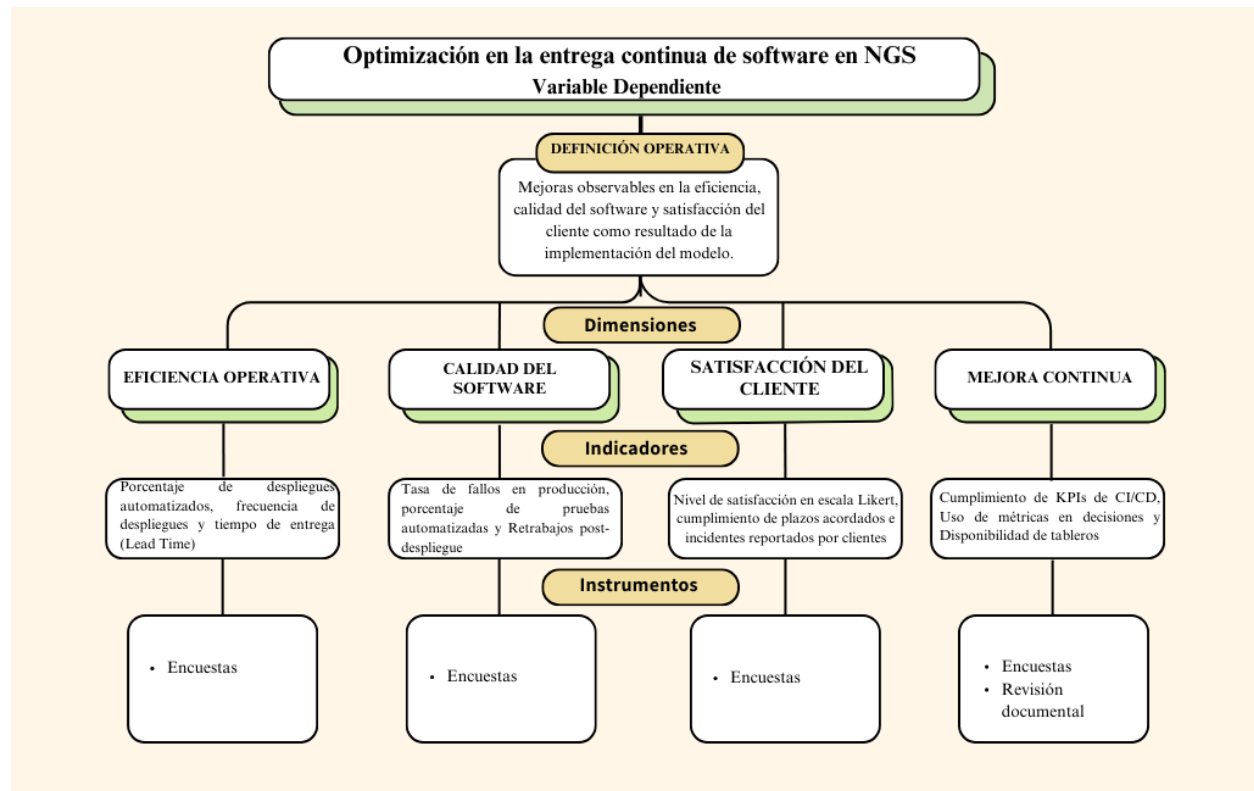
### 3.9.1 ESQUEMA DE VARIABLES DE ESTUDIO

**Figura 17** Esquema de la variable independiente



Nota: Elaboración propia.

**Figura 18** Esquema de la variable dependiente



Nota: Elaboración propia.

### 3.9.2 MATRIZ DE OPERACIONALIZACIÓN DE LAS VARIABLES

**Tabla 17** Matriz de operacionalización de las variables

Variable	Definición operativa	Dimensiones	Indicadores	Instrumentos
Modelo de integración DevOps + Ágil + ITIL V4 (Independiente)	Modelo propuesto que busca estandarizar procesos y prácticas de CI/CD en NGS, incorporando herramientas de automatización y alineación con ITIL V4.	Procesos y brechas actuales.	Número de deficiencias identificadas en CI/CD, Herramientas y competencias.	Encuesta, Análisis FODA, Checklist técnico.
		Integración de prácticas DevOps.	Grado de incorporación de herramientas (Jenkins, Docker, Git, etc.) y políticas estandarizadas.	Encuesta, Revisión documental, Matriz comparativa de herramientas DevOps, Ciclo CI/CD DevOps.

		Alineación con ITIL V4.	Nivel de cumplimiento de prácticas ITIL en gestión de cambios, incidentes y despliegues.	Encuesta, Matriz de correspondencia ITIL, Revisión documental.
		Automatización y Arquitectura tecnológica.	Porcentaje de procesos automatizados (tests, despliegues, monitoreo), Nivel de implementación de pipelines CI/CD, Madurez de la infraestructura tecnológica.	Encuestas, Revisión documental
Optimización en la entrega continua de software en NGS (Dependiente)	Mejoras observables en la eficiencia, calidad del software y satisfacción del cliente como resultado de la implementación del modelo.	Eficiencia operativa.	Porcentaje de despliegues automatizados, frecuencia de despliegues y tiempo de entrega (Lead Time)	Encuesta
		Calidad del software.	Tasa de fallos en producción, porcentaje de pruebas automatizadas y Retrabajos post-despliegue	

		Satisfacción del cliente.	Nivel de satisfacción en escala Likert, cumplimiento de plazos acordados e incidentes reportados por clientes.	
		Mejora continua	Cumplimiento de KPIs de CI/CD, Uso de métricas en decisiones y Disponibilidad de tableros.	Encuestas, Revisión documental.

Nota: Elaboración propia.

### 3.10 TÉCNICAS, INSTRUMENTOS, PROCEDIMIENTOS, ANÁLISIS DE LOS DATOS

#### 3.10.1 TÉCNICAS

Es el método o modalidad general de recolección de información. Las técnicas de investigación son “pasos, medios o formas que se siguen para llevar a cabo una investigación”. En palabras de Campos y Flores (2021), las técnicas son recursos genéricos que acompañan el proceso de investigación, permitiendo la obtención y análisis de datos. Olivos Romero (2023) aclara que “los instrumentos y técnicas de investigación hacen referencia a los procedimientos, y nos enseñan cómo se debe recorrer ese camino”. En el enfoque cualitativo se utilizan técnicas como la observación, la entrevista y la encuesta cualitativa.

En este apartado se muestran las técnicas específicas que usaremos en este estudio:

**Encuesta:** Es una técnica de recolección de datos en la que se aplican cuestionarios con preguntas abiertas. En este caso, se usarán preguntas de este tipo para conocer percepciones y experiencias de los participantes. Según Ranganathan (2023), un cuestionario de investigación puede definirse como una herramienta de recolección de datos compuesta de una serie de preguntas o ítems usados para extraer información de los encuestados sobre sus actitudes, experiencias u opiniones. [Ver anexo 3.](#)

Para nuestro estudio en NGS, se elaborará un cuestionario cualitativo con preguntas

abiertas que estarán alineadas con las dimensiones de análisis.

**Análisis documental:** Este método implica revisar metódicamente documentos y regulaciones significativos, como marcos legales, estándares o documentación técnica (como la relacionada con DevOps o ITIL 4). Su valor radica en su capacidad para obtener información cualitativa de fuentes escritas, lo cual es crucial para estudios cuyo objetivo es comprender las políticas, reglas o directrices institucionales.

Asimismo, puede interpretarse como un procedimiento que permite acceder a la información con el fin de convertir los datos en conocimiento. De acuerdo con Marcelino Aranda et al. (2024), este procedimiento consta de múltiples fases: la búsqueda, la elección, la recolección, la clasificación, la organización, el análisis, la interpretación y, por último, la presentación de los hallazgos.

El análisis documental de esta investigación se centrará en revisar manuales internos de NGS, políticas de TI, estándares pertinentes a DevOps, reportes de proyectos previos y literatura académica vinculada a las metodologías ágiles y DevOps. Esta técnica permitirá la identificación de pruebas respecto a las brechas existentes, las herramientas actuales y los elementos fundamentales para desarrollar un plan que implemente gradualmente DevOps dentro de la empresa. [Ver Anexo 4.](#) [Ver anexo 6.](#) [Ver anexo 2.](#)

**Comparación de instrumentos (Matriz comparativa):** La comparación cualitativa es una técnica analítica que organiza las características de varios elementos de estudio en forma tabular, lo que permite visualizar similitudes y diferencias para identificar criterios de decisión pertinentes (Martínez-Restrepo y Ramírez, 2021).

El método comparativo se aplicará a las herramientas de DevOps (por ejemplo, Jenkins, GitLab, CI/CD, Docker, Kubernetes), así como a las prácticas actuales de NGS, con el fin de analizar su automatización, integración con metodologías ágiles, usabilidad y nivel de adopción. Esta técnica cualitativa ayudará a determinar qué combinaciones de herramientas y prácticas son las más prácticas para integrar en el flujo de CI/CD propuesto. [Ver anexo 5.](#)

**Análisis FODA:** Esta técnica implica la realización de una evaluación sistemática de los componentes internos, identificando tanto las fortalezas como las debilidades de una organización. De la misma forma, se lleva a cabo una evaluación externa que considera las oportunidades y

amenazas presentes en el entorno. Este enfoque permite diagnosticar de manera integral la situación de la organización. Asimismo, se puede considerar como una herramienta accesible que facilita la obtención de una perspectiva general sobre la situación estratégica de una organización específica.

Esto permitirá categorizar:

- Fortalezas: experiencia con técnicas ágiles y habilidades técnicas actuales.
- Debilidades: restricciones de habilidades, resistencia cultural o deficiencias en la automatización.
- Oportunidades: tendencias mundiales en la adopción de DevOps, demanda del mercado y beneficios competitivos.
- Amenazas: la escasez de recursos, los riesgos tecnológicos y la presión de la competencia.

Este análisis de contexto facilitará la conexión entre los resultados cualitativos extraídos de las entrevistas y el análisis de documentos y la metodología recomendada para una implementación gradual. [Ver anexo 1.](#)

### 3.10.2 INSTRUMENTOS

Los instrumentos, en el campo de la investigación, son las herramientas concretas que se crean para implementar con confiabilidad y validez las técnicas de recolección de datos. Hernández-Sampieri, Fernández y Baptista (2014) indican que la objetividad, la validez y la confiabilidad de los datos deben ser aseguradas por cada instrumento, ya que estos elementos constituyen el vínculo práctico entre el marco conceptual de investigación y la adquisición de pruebas empíricas. La correcta elaboración, implementación y codificación de estos instrumentos posibilita la obtención de datos ordenados y exactos para su análisis posterior.

Los instrumentos que utilizaremos se describen a continuación:

**Cuestionario de preguntas abiertas:** Se trata de un instrumento concebido para la obtención de información extensa y detallada de los participantes. De acuerdo con Hernández-Sampieri, Fernández y Baptista (2014), los cuestionarios constituyen una herramienta para la recopilación de datos relacionados con creencias, opiniones, actitudes y comportamientos. Este

instrumento proporciona al encuestado la posibilidad de expresar sus respuestas de manera libre, especialmente en el caso de las preguntas abiertas.

El cuestionario cualitativo permite al investigador explorar a fondo las opiniones de los participantes sobre la integración de DevOps y metodologías ágiles, lo que está en línea con el objetivo de entender sus opiniones y puntos de vista.

**Matriz de análisis documental:** Son herramientas que se han creado para estructurar y sintetizar la información obtenida de fuentes escritas. En la práctica, se crean tablas que organizan datos importantes de documentos internos, normas DevOps, manuales ITIL y otros, con base en categorías establecidas. Estas fichas permiten sistematizar la información adquirida, lo que resulta indispensable para lograr el objetivo de llevar a cabo un análisis crítico del marco de DevOps/ITIL y de la documentación interna de la consultora.

**Matriz comparativa:** Se trata de una tabla de doble entrada que sintetiza datos para contrastar elementos semejantes de acuerdo con criterios particulares.

La matriz comparativa, al ser utilizada en la investigación, sirve para valorar las herramientas DevOps. Cada una de sus líneas representa una herramienta y cada columna, un criterio significativo (por ejemplo, la integración continua, el despliegue automatizado o la compatibilidad con ITIL), lo que permite realizar una comparación sistemática. Esta organización posibilita que se interprete y compare visualmente la manera en la que cada herramienta cumple con los criterios establecidos, lo cual concuerda con el objetivo de seleccionar las opciones más adecuadas para el modelo de integración DevOps-ágil.

**Matriz FODA:** Oña y Vega (2018) explican: “El análisis FODA ha prevalecido en el tiempo como una importante herramienta de planificación y gestión estratégica, facilitando un análisis interno y externo de la organización mediante la cual pueden establecer acciones estratégicas que les permitan alcanzar sus objetivos” (Oña & Vega, 2018, p. 436).

La matriz FODA, en el marco de la tesis, se utiliza para examinar de forma cualitativa la aplicación de DevOps e ITIL en la consultora. Permite detectar dentro de la empresa las limitaciones y las capacidades fundamentales, además de los elementos externos que pueden favorecer o dificultar el éxito. Así, la FODA sistematiza la información cualitativa recopilada (por ejemplo, a través de entrevistas y documentación) y vincula estos elementos con las metas del

estudio, lo que permite el diseño de tácticas adecuadas a la realidad de la organización.

### 3.10.3 PROCEDIMIENTOS

Los procedimientos son una secuencia de pasos sistematizados que se emplean para recopilar y examinar datos en la investigación científica, asegurando así la coherencia y rigurosidad del proceso. Documentan meticulosamente cada paso para garantizar que los resultados sean confiables y constantes, desde la codificación de la información inicial hasta la generación de categorías e hipótesis (Hernández, Fernández y Baptista, 2014).

**Tabla 18** *Relación entre técnicas, instrumentos, procedimientos y objetivos específicos*

<b>Objetivo específico</b>	<b>Técnica</b>	<b>Instrumento</b>	<b>Procedimiento</b>
Analizar qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps.	Encuesta, Análisis FODA, Matriz comparativa.	Cuestionario mixto (escala Likert y preguntas abiertas), Matriz FODA, Checklist técnico.	Diseñar y aplicar el cuestionario en Google Forms al personal técnico de NGS y a un cliente externo; recopilar y organizar las respuestas; codificar la información; clasificar los datos en una matriz FODA para identificar brechas.
Identificar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI.	Encuesta, Revisión documental, Matriz comparativa.	Cuestionario mixto (escala Likert y preguntas abiertas). Matriz de análisis documental, Matriz de correspondencia ITIL, Matriz comparativa de herramientas DevOps.	Aplicar el cuestionario a desarrolladores y personal clave; recopilar documentación interna y lineamientos técnicos si existen; registrar y sistematizar los hallazgos; comparar prácticas actuales con referentes DevOps-Ágil-ITIL V4; analizar convergencias y vacíos para su incorporación.
Evaluar la arquitectura tecnológica más adecuada para soportar la implementación de un flujo de CI/CD en NGS, garantizando la escalabilidad, la seguridad y la trazabilidad.	Encuesta, Análisis documental.	Cuestionario mixto (escala Likert y preguntas abiertas), Matriz de análisis documental.	Recopilar información mediante el cuestionario; reunir documentos técnicos (infraestructura, repositorios, pipelines, políticas de TI); elaborar fichas comparativas; contrastar el estado actual con buenas prácticas de CI/CD; sintetizar hallazgos para valorar la arquitectura disponible.
Proponer un conjunto de indicadores clave de desempeño (KPIs) que permitan evaluar de manera objetiva el impacto de la adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software y satisfacción del cliente.	Encuesta, Análisis documental.	Cuestionario mixto (escala Likert y preguntas abiertas), Matriz de análisis documental.	Analizar la percepción sobre desempeño mediante el cuestionario; identificar variables medibles; derivar KPIs relacionados con eficiencia, calidad y satisfacción del cliente.

Nota: Elaboración propia.

### 3.10.4 ANÁLISIS DE LOS DATOS

El análisis de datos cualitativos es un proceso que se caracteriza por ser sistemático y, a la vez, flexible. En este, los datos obtenidos se agrupan, codifican y clasifican en categorías para detectar patrones y significados que sean pertinentes para el estudio. Este método es esencialmente inductivo y busca la conexión directa entre los hallazgos emergentes y las interrogantes e intenciones de indagación (Hernández, Fernández & Baptista, 2014).

A continuación, se muestra un plan de análisis para la presente investigación:

**Tabla 19** *Plan de análisis cualitativo de los datos*

<b>Fase</b>	<b>Procedimiento</b>	<b>Resultado esperado</b>
<b>Organización</b>	Reunir y clasificar la información obtenida de entrevistas, encuestas y documentos.	Datos listos y estructurados para su análisis.
<b>Codificación</b>	Identificar fragmentos clave y asignar códigos simples.	Lista organizada de códigos iniciales.
<b>Categorización</b>	Agrupar los códigos en temas principales (procesos, herramientas, competencias, ITIL V4, KPIs).	Conjunto de categorías temáticas.
<b>Interpretación</b>	Relacionar categorías con los objetivos; elaborar matrices comparativas y FODA; derivar propuestas de mejora y KPIs.	Conclusiones alineadas con los objetivos de la investigación.

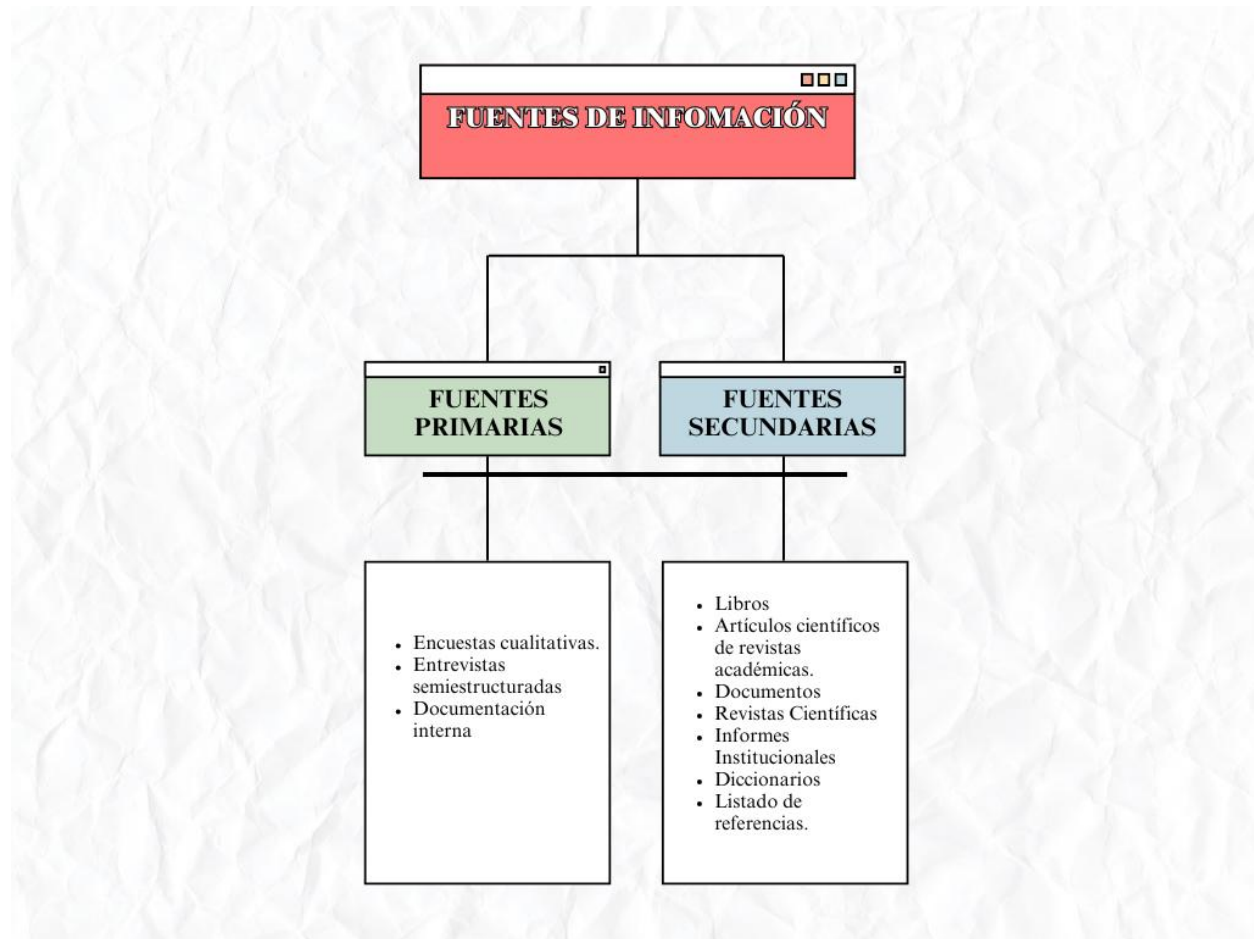
Nota: Elaboración propia basada en un enfoque cualitativo.

### 3.11 FUENTES DE INFORMACIÓN

Las fuentes de información son el fundamento para la elaboración de cualquier investigación, pues facilitan el acceso a datos relevantes y fiables que permiten responder a los objetivos propuestos. En este sentido, Arias (2012) afirma que “las fuentes de información son los diversos documentos, personas u objetos de donde se obtiene la información necesaria para llevar a cabo una investigación” (p. 56). En este enfoque, se distinguen las fuentes primarias que nos proporcionan información directa y original obtenida mientras se trabaja en el campo de las fuentes secundarias, que ofrecen apoyo teórico y contextual a través del uso de estudios, libros, artículos

y documentos que ya han sido desarrollados.

**Figura 19** Clasificación de las fuentes de la información



Nota: Elaboración propia.

### 3.11.1 FUENTES PRIMARIAS

La información obtenida en el trabajo de campo será la fuente principal de esta investigación y estará estrechamente relacionada con las metas concretas del estudio. Primeramente, se realizarán encuestas cualitativas a los colaboradores de la Consultora Nueva Generación de Software (NGS) utilizando cuestionarios abiertos. Estos cuestionarios se diseñan con el fin de obtener perspectivas sobre las brechas en herramientas, procesos y competencias que afectarán la implementación de prácticas DevOps. En segundo lugar, se llevarán a cabo entrevistas semiestructuradas con personal fundamental de la organización, como líderes de proyecto, expertos en QA, desarrolladores, especialistas en DevOps y empleados de operaciones. El propósito es profundizar en los retos, las prácticas actuales y la viabilidad de poner en práctica metodologías ágiles y DevOps dentro del marco organizacional que ya existe. Esto posibilitará

una visión precisa y pormenorizada de los procedimientos y prácticas institucionales.

Por último, se examinarán los documentos internos de la consultora, incluidos los manuales de procesos, las directrices de proyectos y las políticas de calidad. Este material será útil para entender de manera clara y práctica cómo se trabaja dentro de la organización, proporcionando una visión directa de sus procesos y de las dinámicas que ya son parte de su cultura institucional. Así, las fuentes primarias en su totalidad proporcionarán información verídica, que no solo muestre los procesos formales, sino también las vivencias y puntos de vista de los individuos que participan en ellos cotidianamente.

### 3.11.2 FUENTES SECUNDARIAS

Los medios secundarios empleados en este estudio comprenden un extenso rango de materiales bibliográficos y documentales que funcionan como apoyo teórico y conceptual.

Entre ellas se encuentran libros enfocados en DevOps y metodologías de investigación, así como la administración de las tecnologías informáticas y marcos de trabajo ágiles. Por ejemplo, los aportes de Van Bon (2019) y Hernández-Sampieri y Mendoza (2023), que brindan principios prácticos y metodológicos fundamentales para el análisis. Se tomaron en cuenta, además, tesis y trabajos académicos que analizan el uso de ITIL 4, DevOps y Scrum en varios sectores como la banca, la industria, el sector gubernamental y la educación; esto posibilita establecer comparaciones con el caso objeto de estudio. También se revisaron publicaciones en revistas científicas y portales especializados, como Dialnet, la Revista Española de Ingeniería de Software y Journal of Information Technology Service Management, que ofrecen información reciente acerca de tendencias y prácticas eficaces. Por último, se agregaron reportes institucionales y fuentes internacionales que examinan el impacto del nearshoring, las tendencias globales en tecnología y la legislación vigente para el sector de TI en Honduras y la región.

### 3.12 MATRIZ METODOLÓGICA

Tabla 20 Matriz de congruencia

Preguntas de investigación	Objetivos específicos	Metodología	Variable	Dimensiones	Indicadores	Instrumentos
¿Qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps?	Analizar qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps.	Mixta con predominio cualitativo	Modelo de integración DevOps + Ágil + ITIL V4 (Independiente)	Procesos y brechas actuales.	Número de deficiencias identificadas en CI/CD, Herramientas y competencias.	Encuestas, Checklist técnico, Análisis FODA.
¿Qué prácticas, herramientas y políticas DevOps puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI?	Identificar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI.			Integración de prácticas DevOps.	Grado de incorporación de herramientas (Jenkins, Docker, Git, etc.) y políticas estandarizadas.	Encuesta, Revisión documental, Matriz comparativa de herramientas DevOps, Ciclo CI/CD DevOps.
				Alineación con ITIL V4.	Nivel de cumplimiento de prácticas ITIL en gestión de cambios, incidentes y despliegues.	Encuesta, Matriz de correspondencia a ITIL, Revisión documental.

<p>¿Qué arquitectura tecnológica es más adecuada para soportar la implementación de CI/CD en NGS, garantizando escalabilidad, seguridad y trazabilidad?</p>	<p>Evaluar la arquitectura tecnológica más adecuada para soportar la implementación de un flujo de CI/CD en NGS, garantizando la escalabilidad, la seguridad y la trazabilidad.</p>
<p>¿Qué indicadores clave de desempeño (KPIs) deben definirse para medir el impacto de adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software, productividad y la satisfacción del cliente?</p>	<p>Proponer un conjunto de indicadores clave de desempeño (KPIs) que permitan evaluar de manera objetiva el impacto de la adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software y satisfacción del cliente.</p>

<p>Optimización en la entrega continua de software en NGS (Dependiente)</p>	<p>Automatización y Arquitectura tecnológica.</p>	<p>Porcentaje de procesos automatizados (tests, despliegues, monitoreo), Nivel de implementación de pipelines CI/CD, Madurez de la infraestructura tecnológica.</p>	<p>Encuestas, Revisión documental.</p>
	<p>Eficiencia operativa.</p>	<p>Porcentaje de despliegues automatizados, frecuencia de despliegues y tiempo de entrega (Lead Time).</p>	
	<p>Calidad del software.</p>	<p>Tasa de fallos en producción, porcentaje de pruebas automatizadas y Retrabajos post-despliegue.</p>	<p>Encuesta</p>
	<p>Satisfacción del cliente.</p>	<p>Nivel de satisfacción en escala Likert, cumplimiento de plazos acordados e incidentes reportados por clientes.</p>	
<p>Mejora continua</p>	<p>Cumplimiento de KPIs de CI/CD, Uso de métricas en</p>	<p>Encuestas, Revisión documental.</p>	

					decisiones y Disponibilidad de tableros.	
--	--	--	--	--	--	--

Nota: Elaboración propia.

Para resumir la matriz de congruencia, a continuación, te dejo un mapa que relaciona las variables, dimensiones y objetivos.

**Figura 20** Mapa de relación entre variables, dimensiones y objetivos

Mapa de Relación entre Variables, Dimensiones y Objetivos					
Variables	Dimensiones	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4
Modelo de integración DevOps + Ágil + ITIL V4	Procesos y brechas actuales	✓	✗	✗	✗
	Integración de prácticas DevOps	✗	✓	✗	✗
	Alineación con ITIL V4	✗	✓	✗	✗
	Automatización y arquitectura	✗	✗	✓	✗
Optimización en la entrega continua de software en NGS	Eficiencia operativa	✗	✗	✗	✓
	Calidad del software	✗	✗	✗	✓
	Satisfacción del cliente	✗	✗	✗	✓
	Mejora continua	✗	✗	✗	✓

Nota: Elaboración propia.

## CAPÍTULO IV - RESULTADOS Y ANÁLISIS

En este capítulo se presenta el análisis de la situación actual de la Consultora Nueva Generación de Software (NGS) en relación con sus procesos de desarrollo y entrega de software, el uso de prácticas DevOps y ágiles, el grado de alineación con ITIL V4, la arquitectura tecnológica disponible y la gestión de indicadores de desempeño. El propósito es ofrecer una visión ordenada y comprensible del estado actual de la organización, tomando como referencia los objetivos específicos planteados en este trabajo.

A partir de esta revisión se identifican las principales brechas, fortalezas y oportunidades de mejora que influyen en la implementación de un flujo de Integración y Entrega Continua (CI/CD) en NGS. Estos hallazgos servirán como base para la formulación del modelo de integración DevOps + Ágil + ITIL V4 y para las propuestas de mejora que se desarrollan en los capítulos posteriores.

### 4.1 INSTRUMENTOS Y PARTICIPANTES

En esta sección se enlistan de manera general los instrumentos que fueron aplicados para los objetivos específicos, se describe cómo fueron aplicados y su validación, y también se detallan algunas características de los participantes en cada uno de los instrumentos. A continuación, se describe cada uno de ellos:

#### **ENCUESTA PRINCIPAL PARA LA EVALUACIÓN DE PROCESOS, ARQUITECTURA Y PRÁCTICAS DE DESARROLLO EN NGS**

La encuesta diseñada para esta investigación se aplicó en la Consultora Nueva Generación de Software (NGS) mediante un formulario elaborado en Google Forms. El instrumento fue compartido de manera formal a los colaboradores de la empresa, contando con el respaldo institucional para su distribución y participación. Su propósito fue recopilar información sobre los procesos, herramientas, prácticas tecnológicas, arquitectura de desarrollo y mecanismos de medición existentes en NGS, con el fin de evaluar la situación actual y responder a los objetivos específicos planteados en el estudio.

El cuestionario fue estructurado en secciones diferenciadas, cada una alineada a un objetivo específico de la investigación:

**Sección de datos demográficos:** Diseñada para caracterizar la muestra participante según

rol, experiencia y área de trabajo.

**Sección A (Procesos y brechas actuales):** Relacionada con el Objetivo 1, orientada a identificar limitaciones en documentación, roles, flujos de trabajo y automatización.

**Sección B (Prácticas DevOps aplicables y alineación con ITIL V4):** Dirigida al Objetivo 2, para conocer el nivel de adopción de herramientas, políticas y prácticas DevOps que podrían integrarse al marco ágil existente.

**Sección C (automatización y arquitectura tecnológica):** Asociada al Objetivo 3, enfocada en evaluar la madurez de la infraestructura, herramientas y mecanismos de soporte para un flujo CI/CD seguro y escalable.

**Sección D y E (KPIs, métricas e impacto en la entrega continua):** Vinculadas al Objetivo 4, orientadas a identificar el uso actual de indicadores y la percepción del impacto en tiempos de entrega, calidad y satisfacción del cliente.

Como parte del proceso metodológico, la encuesta fue validada mediante una prueba piloto aplicada a cinco desarrolladores de NGS, seleccionados por su experiencia y conocimiento del ciclo de desarrollo de software. Los participantes de la prueba piloto brindaron observaciones y sugerencias que permitieron mejorar la redacción de las preguntas, aclarar escalas de respuesta y ajustar la secuencia del instrumento. Estos validadores formaron parte oficialmente de la muestra final del estudio.

Una vez validada, la encuesta fue distribuida al personal seleccionado, obteniéndose un total de 26 respuestas. De ellas, 23 correspondieron a colaboradores de NGS pertenecientes a distintas áreas de desarrollo, QA y la gerencia. Las 3 respuestas restantes provinieron de profesionales externos pertenecientes a un cliente de NGS, quienes participaron debido a su vínculo operativo con la empresa y su experiencia en roles clave como QA, Gestión de Proyectos y Product Owner. Su inclusión permitió complementar la visión interna con la perspectiva de actores que interactúan directamente con los procesos de desarrollo de software de NGS.

## **DESCRIPCIÓN DE LA APLICACIÓN DEL INSTRUMENTO FODA**

El análisis FODA se aplicó como un instrumento temático elaborado a partir de la percepción, observación y experiencia directa de los investigadores dentro del ciclo de vida de desarrollo de software en NGS. Su construcción se basó en la participación activa en proyectos y

en la interacción constante con los diferentes equipos técnicos, lo que permitió identificar de manera cualitativa las fortalezas, oportunidades, debilidades y amenazas presentes en los procesos actuales. Este instrumento permitió captar aspectos operativos que no siempre se evidencian por medio de encuestas formales, complementando así la recolección de datos obtenida por otros instrumentos.

### **DESCRIPCIÓN DEL CHECKLIST TÉCNICO APLICADO PARA LAS HERRAMIENTAS, POLÍTICAS Y PRÁCTICAS EN NGS**

El Checklist técnico se aplicó como un instrumento estructurado para evaluar el nivel de adopción y madurez de las herramientas, políticas y prácticas relacionadas con la integración continua, entrega continua y automatización dentro de NGS. Su diseño se elaboró a partir de indicadores clave que reflejan el uso real de tecnologías como Jenkins, Docker, Git, pipelines automatizados y políticas formales de versionado, entre otros aspectos críticos del flujo CI/CD. Este instrumento fue aplicado al Gerente general de NGS en conjunto con los investigadores, permitiendo así una valoración directa de la situación actual desde la perspectiva de la dirección, identificándose niveles de capacidad mayoritariamente bajos y medios en áreas esenciales como la automatización de compilaciones, pruebas y despliegues, así como en la formalización de políticas de control de versiones y manejo de entornos.

### **DESCRIPCIÓN DE LA MATRIZ DE MADUREZ DE PRÁCTICAS ITIL V4 EN NGS**

Para evaluar el nivel actual de adopción de las prácticas ITIL V4 en la Consultora NGS, se aplicó una Matriz de Madurez basada en el ITIL Maturity Model (ITIL MM) de PeopleCert, modelo descrito previamente en el capítulo II. Esta matriz fue adaptada por los investigadores seleccionando únicamente las prácticas ITIL consideradas más relevantes para su alineación con DevOps y con las necesidades operativas de NGS, tales como: Gestión de Cambios, Gestión de Despliegues, Gestión de Incidentes, Gestión de Problemas, Mejora Continua, Monitoreo y Gestión de Eventos, así como Gestión de Configuración y Activos.

Debido a que la mayoría del personal se encontraba trabajando bajo la modalidad de teletrabajo, el instrumento se aplicó de manera online, utilizando un cuestionario digital dividido por secciones, donde cada práctica ITIL se evaluó independientemente. Para cada práctica, los participantes debían seleccionar el nivel de madurez actual de NGS, utilizando una escala del 0 al 5 tomada del ITIL Maturity Model, y posteriormente brindar una justificación cualitativa sobre su percepción del nivel seleccionado. Esta estructura permitió recopilar información precisa, tanto

cuantitativa como descriptiva, garantizando profundidad en las valoraciones sin dificultar la participación de los colaboradores.

En total se obtuvieron 24 respuestas válidas, provenientes de desarrolladores, QA/Testers y la gerencia. La diversidad de participantes permitió obtener una visión más completa del estado actual de cada práctica y de las diferencias en su implementación según el tipo de proyecto o equipo. La información recolectada a través de esta matriz constituye la base para el análisis detallado del nivel de madurez ITIL en la organización y para la identificación de brechas críticas que deben atenderse para fortalecer la integración con DevOps y mejorar la gestión de servicios de TI.

### **DESCRIPCIÓN DE LA REVISIÓN DOCUMENTAL EN NGS**

La revisión documental permitió sustentar teórica y técnicamente el análisis realizado en la investigación, mediante el estudio sistemático de estándares internacionales, marcos de referencia y buenas prácticas relacionadas con DevOps, metodologías ágiles, ITIL V4 y la gestión del desempeño en entornos de entrega continua.

A través del análisis de la literatura especializada, se identificaron prácticas, herramientas y políticas que permiten fortalecer los procesos de desarrollo, pruebas, despliegue y operación del software, así como su adecuada alineación con principios de gobernanza y gestión de servicios. De igual forma, la revisión permitió establecer criterios técnicos para definir arquitecturas tecnológicas escalables, seguras y trazables que soporten de manera eficiente el flujo de integración y entrega continua.

### **DESCRIPCIÓN DE LA MATRIZ COMPARATIVA EN NGS**

La Matriz Comparativa fue utilizada como un instrumento de análisis para contrastar el estado actual de los procesos, herramientas y prácticas utilizadas en la consultora Nueva Generación de Software (NGS) frente a modelos de referencia y buenas prácticas reconocidas en DevOps, metodologías ágiles y el marco de gestión de servicios ITIL V4.

Este instrumento permitió identificar similitudes, diferencias y brechas entre la situación real de NGS y los enfoques recomendados a nivel internacional, facilitando una evaluación objetiva del nivel de alineación de la organización con prácticas estandarizadas de integración y entrega continua (CI/CD). La matriz consideró aspectos clave como automatización, control de versiones, gestión de cambios, monitoreo, trazabilidad, documentación y uso de métricas.

A continuación, se muestra en una figura el resumen de los instrumentos que se aplicaron por cada uno de los objetivos:

**Figura 21** Relación de los objetivos específicos por cada instrumento aplicado

MAPA CONCEPTUAL DE RELACION DE OBJETIVOS CON INSTRUMENTOS APLICADOS				
INSTRUMENTO UTILIZADO	OBJETIVO 1	OBJETIVO 2	OBJETIVO 3	OBJETIVO 4
	Analizar qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps.	Identificar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI.	Evaluar la arquitectura tecnológica más adecuada para soportar la implementación de un flujo de CI/CD en NGS, garantizando la escalabilidad, la seguridad y la trazabilidad.	Proponer un conjunto de indicadores clave de desempeño (KPIs) que permitan evaluar de manera objetiva el impacto de la adopción de DevOps en NGS, considerando los tiempos de entrega, calidad del software y satisfacción del cliente.
ENCUESTA	✓	✓	✓	✓
ANÁLISIS FODA	✓	X	X	X
CHECKLIST LIST TÉCNICO	✓	X	X	X
REVISIÓN DOCUMENTAL	X	✓	✓	✓
MATRIZ DE CORRESPONDENCIA ITIL	X	✓	X	X
MATRIZ COMPARATIVA DE HERRAMIENTAS DEVOPS	X	✓	X	X

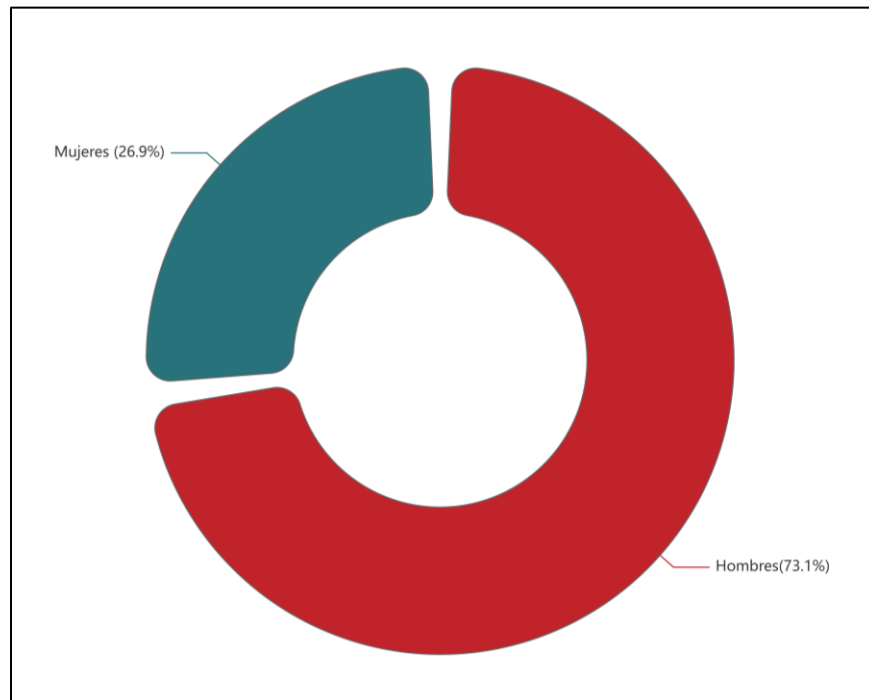
Nota: Elaboración propia.

## 4.2 DESCRIPCIÓN DE LOS RESULTADOS Y SU ANÁLISIS POR OBJETIVO

En esta sección se presentan los resultados obtenidos mediante los distintos instrumentos utilizados en la investigación, organizados de acuerdo con cada uno de los objetivos específicos. Se describe de forma clara la información recopilada y se lleva a cabo un análisis general que permite identificar las tendencias principales observadas en los procesos y prácticas actuales de NGS.

A continuación, se presenta la sección de datos demográficos, donde se muestran las características de los participantes de la encuesta a través de gráficos con su respectivo detalle:

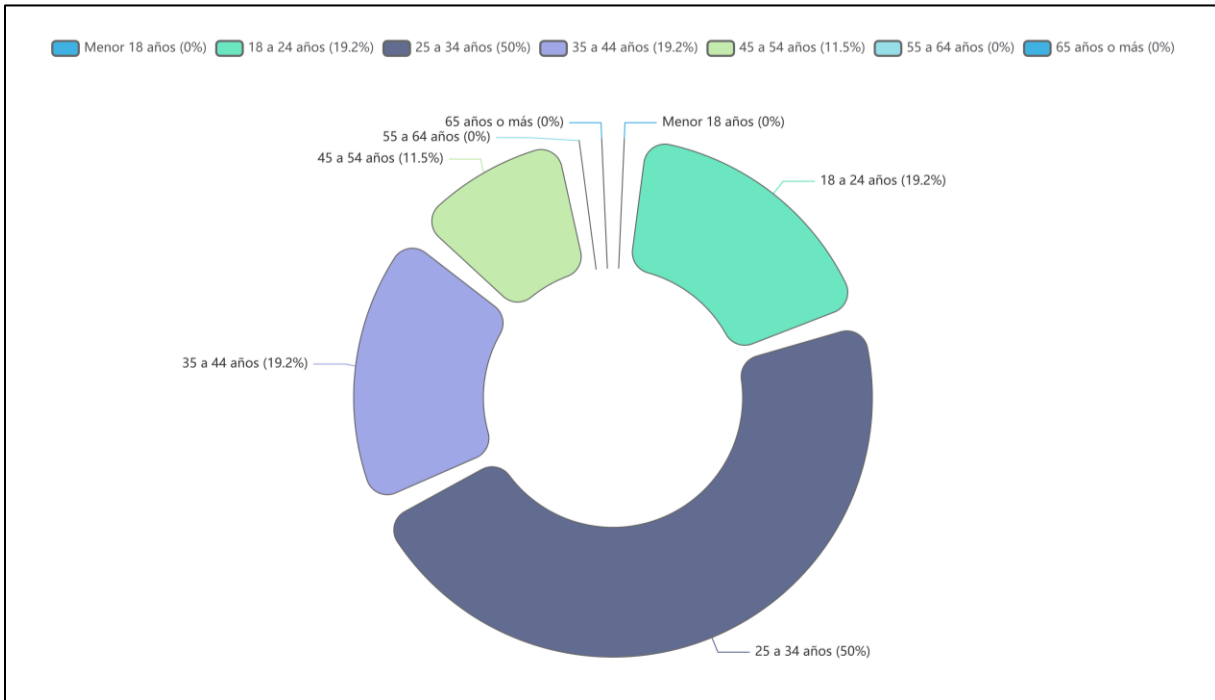
**Figura 22** *Distribución por género del personal encuestado en NGS*



Nota: Elaboración propia.

La muestra participante del estudio estuvo conformada por colaboradores de NGS que forman parte de las áreas de Desarrollo, QA y Operaciones, quienes aportaron información clave para evaluar el estado actual de los procesos, herramientas y arquitectura tecnológica en relación con la adopción de prácticas DevOps y CI/CD. En cuanto a la distribución por género, el 73.1% de los participantes fueron hombres, mientras que el 26.9% correspondió a mujeres. Esta composición refleja una mayor presencia masculina en los equipos tecnológicos de la organización, un aspecto común en entornos de desarrollo de software, y que resulta relevante para contextualizar los resultados obtenidos en los diferentes instrumentos aplicados.

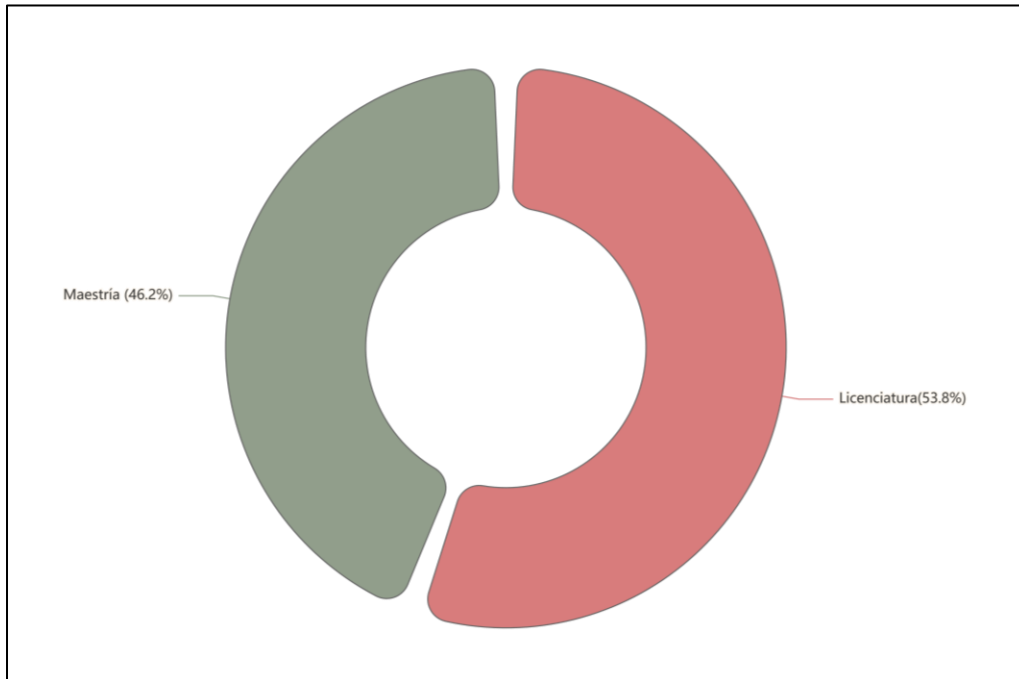
**Figura 23** Distribución por rangos de edad de los participantes



Nota: Elaboración propia.

La distribución por edad de los participantes refleja una muestra predominantemente joven y en etapa activa de desarrollo profesional. El grupo más representativo corresponde a personas entre 25 y 34 años, quienes constituyen el 50% del total de encuestados. Este segmento suele concentrar a profesionales con experiencia intermedia y alto involucramiento en procesos tecnológicos, lo cual aporta una perspectiva relevante para el análisis de nuestro estudio. Asimismo, un 19.2% de los participantes se encuentra en el rango de 18 a 24 años, y otro 19.2% pertenece al grupo de 35 a 44 años, evidenciando la presencia equilibrada de talento joven en formación y profesionales con mayor trayectoria. Finalmente, el 11.5% corresponde a personas entre 45 y 54 años, mientras que no se registraron participantes menores de 18 años ni mayores de 55 años. Esta distribución de edades indica que los resultados del estudio se basan en la opinión de colaboradores con mucha actividad operativa y experiencia importante en desarrollo, pruebas, operaciones y en nuevas prácticas tecnológicas como DevOps.

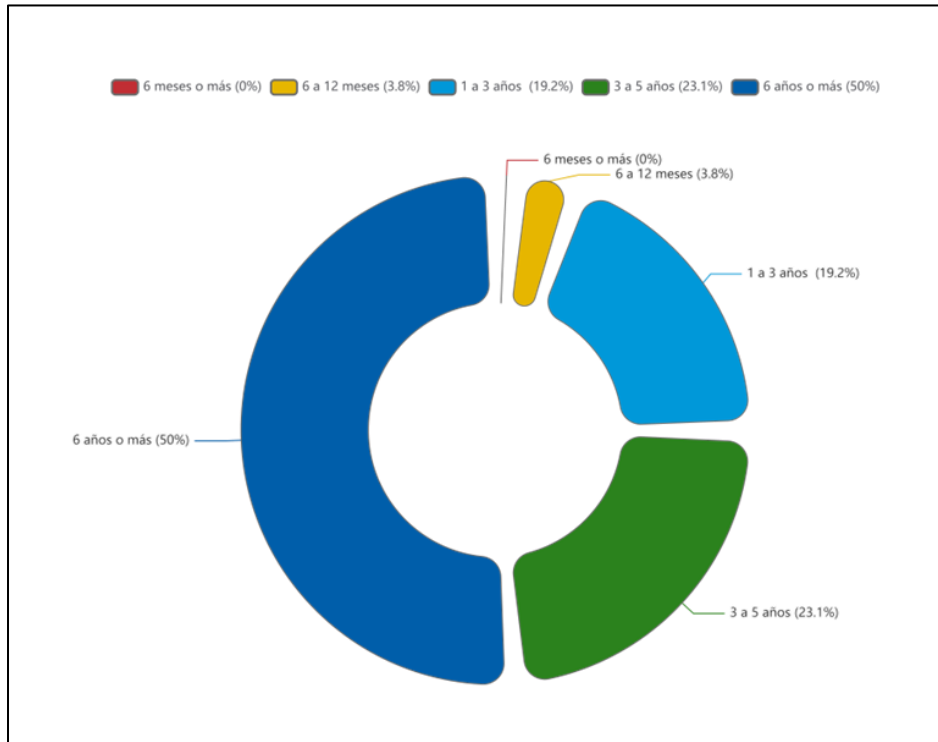
**Figura 24** Nivel educativo alcanzado por los colaboradores participantes



Nota: Elaboración propia.

El nivel educativo de la muestra evidencia una composición equilibrada entre profesionales con formación de pregrado y posgrado. Del total de encuestados, el 53.8% cuenta con estudios de licenciatura, mientras que el 46.2% posee formación a nivel de maestría. Esta distribución indica que una gran parte del personal en Desarrollo, QA y Operaciones tiene un grado académico avanzado. Esto ayuda a mejorar sus habilidades analíticas, técnicas y su adaptación a prácticas modernas como DevOps, CI/CD e ITIL V4. Esta combinación de perfiles profesionales permite obtener una visión completa y fundamentada sobre los procesos tecnológicos evaluados en el estudio.

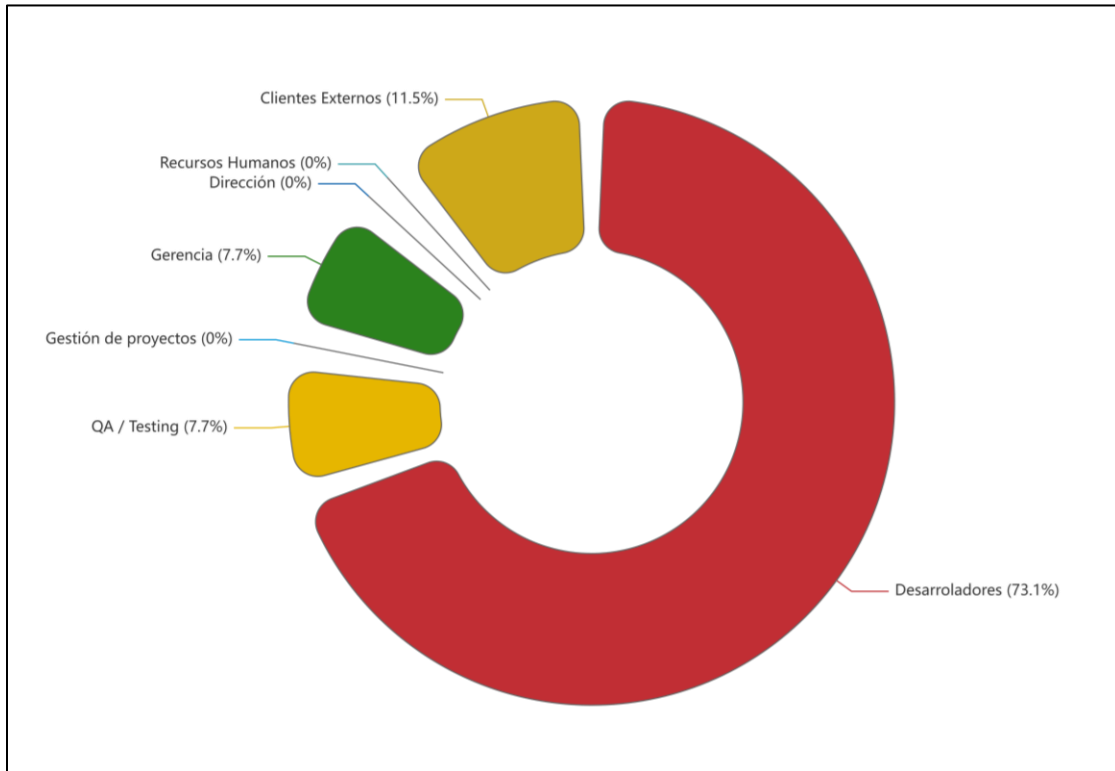
**Figura 25** Años de experiencia en el área de TI o desarrollo del personal encuestado



Nota: Elaboración propia.

La distribución de años de experiencia de los participantes muestra que la muestra está conformada principalmente por personal con un nivel de trayectoria laboral sólido dentro del ámbito tecnológico. El 50% de los encuestados cuenta con 6 años o más de experiencia, lo que evidencia una base profesional madura y con amplio conocimiento práctico sobre procesos de desarrollo, pruebas y operaciones, un 23.1% posee entre 3 y 5 años de experiencia, mientras que el 19.2% tiene entre 1 y 3 años, lo que complementa la muestra con perfiles en crecimiento que también aportan perspectivas actualizadas sobre herramientas modernas y metodologías emergentes, Finalmente, un 3.8% de los participantes reporta entre 6 y 12 meses de experiencia, y no se registraron casos con menos de 6 meses. En conjunto, estos datos reflejan que la mayoría de la muestra cuenta con experiencia suficiente para evaluar de manera crítica el estado actual de la arquitectura tecnológica, las prácticas DevOps y los procesos.

**Figura 26** Cargo o rol desempeñado por los encuestados dentro de NGS

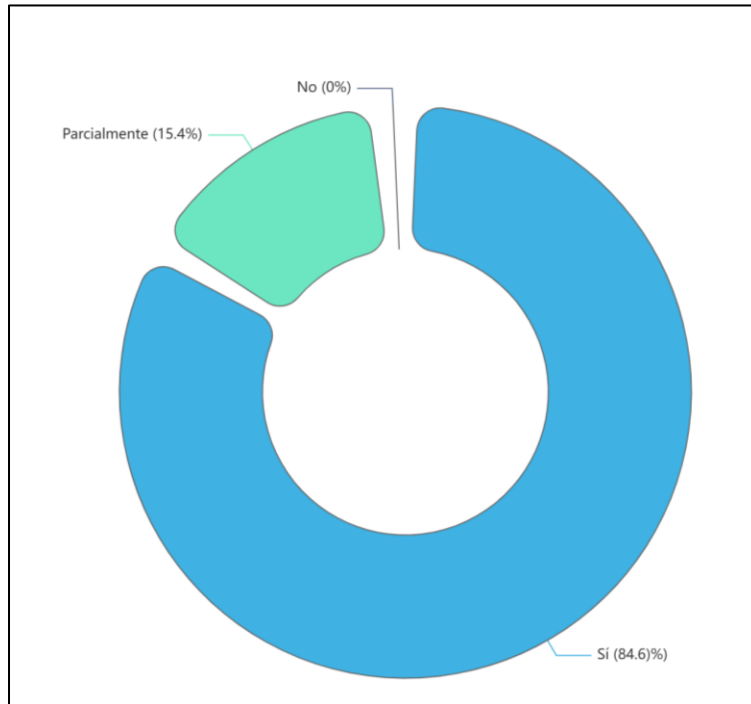


Nota: Elaboración propia.

La distribución de cargos muestra que la mayoría de los participantes pertenece al área de Desarrollo. En concreto, el 73.1% de los encuestados son Desarrolladores, lo que evidencia que la muestra está compuesta principalmente por personal directamente involucrado en la construcción de software y en la ejecución de actividades clave del flujo de los proyectos, también se cuenta con la participación de colaboradores de QA/Testing (7.7%) y Gerencia (7.7%), lo que aporta perspectivas complementarias relacionadas con aseguramiento de la calidad y toma de decisiones operativas. Finalmente, un 11.5% corresponde a Clientes Externos, quienes ofrecen una visión desde la experiencia del usuario final y los requerimientos funcionales.

No se registró participación de personal de Dirección, Gestión de Proyectos ni Recursos Humanos. En conjunto, esta distribución permite interpretar los resultados desde una perspectiva centrada en los roles técnicos operativos.

**Figura 27** Grado de familiaridad de los participantes con el término DevOps

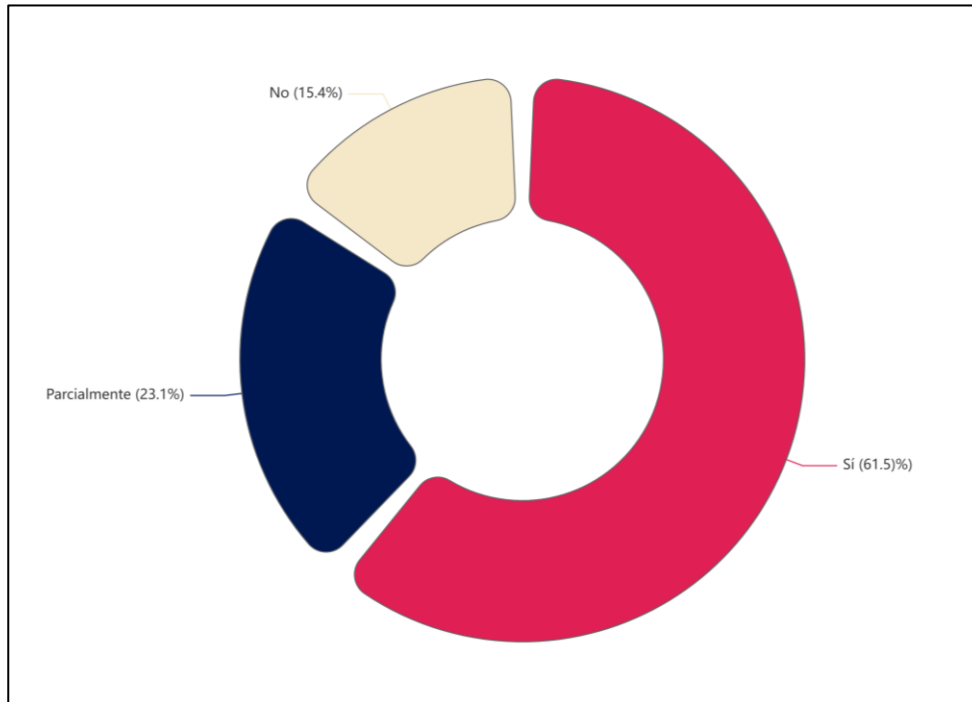


Nota: Elaboración propia.

Los resultados muestran un alto nivel de familiaridad del personal con el concepto de DevOps. El 84.6% de los participantes indicó que sí conoce DevOps, lo que evidencia que la mayoría posee al menos una comprensión general de sus principios, prácticas y beneficios dentro del ciclo de desarrollo y operación. Un 15.4% manifestó conocerlo parcialmente, lo cual sugiere la necesidad de reforzar la formación en conceptos más avanzados, herramientas y flujos de automatización. Finalmente, todos los participantes conocían algo del tema, lo que demuestra que no hay desconocimiento total en la muestra.

Esta distribución es relevante para el estudio, ya que indica que existe una base conceptual favorable para la adopción o fortalecimiento de prácticas DevOps en NGS.

**Figura 28** Grado de familiaridad de los participantes con el marco ITIL V4



Nota: Elaboración propia.

La familiaridad con ITIL entre los participantes muestra un nivel intermedio de conocimiento sobre este marco de gestión de servicios de TI. Del total de encuestados, el 61.5% indicó que sí conoce ITIL, lo que refleja que más de la mitad del personal está familiarizado con sus principios y prácticas fundamentales, como gestión de incidencias, cambios y problemas. Un 23.1% manifestó conocerlo parcialmente, lo cual sugiere que existe experiencia básica, pero con brechas en la comprensión de sus procesos y alineación formal. Finalmente, el 15.4% expresó no conocer ITIL, evidenciando que aún existe un segmento del equipo que no ha tenido exposición directa al marco.

Esta distribución es relevante para la investigación, ya que ITIL V4 constituye un elemento clave para estandarizar procesos, mejorar la trazabilidad y fortalecer la integración con prácticas DevOps. Aunque la mayoría conoce el marco, algunos participantes tienen poco o ningún conocimiento. Esto muestra la necesidad de capacitación interna para lograr una adopción más uniforme y alineada con los objetivos de mejora continua y madurez operativa de NGS.

#### 4.2.1 ANÁLISIS DE LAS BRECHAS EN PROCESOS, HERRAMIENTAS Y COMPETENCIAS ACTUALES EN NGS

A continuación, presentamos los resultados y el análisis de los instrumentos para el objetivo 1:

##### **FORTALEZAS, OPORTUNIDADES, DEBILIDADES Y AMENAZAS (FODA) EN NGS:**

En el análisis de las brechas existentes en los procesos, herramientas y competencias actuales de NGS, se aplicó un análisis FODA que permitió identificar los factores estratégicos que influyen en el desempeño del equipo de desarrollo y en la madurez tecnológica de la organización. Este instrumento proporcionó una visión estructurada de las fortalezas y debilidades internas, así como de las oportunidades y amenazas del entorno, permitiendo comprender cómo estos elementos condicionan la implementación de prácticas de automatización, estandarización y mejoras en la gestión del ciclo de vida del software.

**Figura 29** *Análisis FODA de Procesos, herramientas y competencias actuales en NGS*

<b>Resultados del análisis FODA</b>	
<p><b>Fortalezas</b></p> <ul style="list-style-type: none"> <li>• Personal técnico capacitado en diversas tecnologías y con experiencia en entornos de desarrollo complejos.</li> <li>• Uso de herramientas como Azure DevOps, sistemas de versionamiento y repositorios de código.</li> <li>• Algunos equipos de desarrollo cuentan con experiencia previa en metodologías ágiles y prácticas DevOps.</li> <li>• Uso de tableros con enfoque Kanban para la planificación y seguimiento de tareas diarias.</li> </ul>	<p><b>Debilidades</b></p> <ul style="list-style-type: none"> <li>• Falta de documentación oficial de los procesos operativos y de entrega de software.</li> <li>• Roles poco definidos dentro del marco ágil, lo que genera ambigüedad en responsabilidades.</li> <li>• Desaprovechamiento de los programas de capacitación ofrecidos por la empresa</li> <li>• Falta de indicadores o KPIs para medir la eficiencia, calidad o cumplimiento de entregas.</li> <li>• No todos los desarrolladores poseen madurez suficiente en herramientas de automatización y entrega continua.</li> </ul>
<p><b>Oportunidades</b></p> <ul style="list-style-type: none"> <li>• Existencia de programas internos de capacitación continua a través de plataformas educativas.</li> <li>• Interés institucional en fortalecer la integración de prácticas DevOps y metodologías ágiles.</li> <li>• Posibilidad de implementar herramientas de automatización como Docker y Kubernetes a nivel corporativo.</li> <li>• Potencial de estandarizar procesos y roles mediante la adopción formal de marcos ágiles e ITIL V4</li> </ul>	<p><b>Amenazas</b></p> <ul style="list-style-type: none"> <li>• Resistencia al cambio por parte de algunos colaboradores ante la estandarización y automatización.</li> <li>• La carga laboral y la falta de tiempo disponible pueden limitar la innovación y adopción de nuevas prácticas.</li> <li>• Riesgo de retrasos o errores en despliegues por falta de una arquitectura CI/CD consolidada.</li> <li>• Incremento de la exigencia del entorno tecnológico y de los clientes respecto a la calidad y tiempos de entrega.</li> <li>• La ausencia de procesos estandarizados puede afectar la trazabilidad y el control de versiones.</li> </ul>

Nota: Elaboración propia.

Los resultados del análisis FODA muestran una combinación de aspectos positivos y áreas que requieren atención. Entre las fortalezas identificadas destacan la experiencia técnica del personal, el uso de herramientas modernas como Azure DevOps y sistemas de versionamiento, así como la disposición para adoptar nuevas prácticas. En contraste, se evidenciaron debilidades relacionadas con la falta de documentación estandarizada, roles poco definidos dentro del marco ágil, dependencia de conocimientos individuales y bajo aprovechamiento de las instancias de capacitación. De igual forma, se reconocieron oportunidades asociadas al interés institucional por fortalecer DevOps y mejorar la infraestructura, mientras que las amenazas surgieron vinculadas a la resistencia al cambio y la carga laboral que limita la innovación.

NGS posee un potencial considerable para avanzar hacia un modelo más automatizado y estandarizado. No obstante, se evidenciaron brechas que requieren atención inmediata. Entre los principales hallazgos se destacan:

- La necesidad de formalizar y documentar procesos.
- La definición clara de roles y responsabilidades.
- El establecimiento de métricas que permitan medir la eficiencia del desarrollo y la entrega de software.
- El fortalecimiento de la adopción de herramientas de automatización y prácticas CI/CD.

A pesar de estas limitaciones, las fortalezas y oportunidades identificadas muestran que existen condiciones favorables para avanzar hacia un modelo integrado basado en DevOps, metodologías ágiles e ITIL V4. El FODA aporta un panorama claro sobre las áreas que requieren intervención y constituye un insumo clave para orientar la propuesta de mejora desarrollada en capítulos posteriores.

### **CHECKLIST TÉCNICO EN NGS:**

Luego del análisis FODA, fue necesario complementar dicha perspectiva con una evaluación más específica y operativa. Para ello, se aplicó un Checklist técnico orientado a medir el nivel de madurez en herramientas, políticas y prácticas asociadas al ciclo de vida del software, especialmente aquellas vinculadas a la integración continua, entrega continua y automatización.

**Tabla 21** Checklist de las herramientas, políticas y prácticas en NGS

NO	Herramienta / política / práctica	Nivel de Capacidad (Bajo, Medio, Alto)	Observación
1	Uso habitual de Jenkins para integración continua	<b>Bajo</b>	El uso de Jenkins es nulo.
2	Uso habitual de Docker para contenedores o despliegues	<b>Medio</b>	Existe conocimiento básico de Docker y se han realizado pruebas piloto en proyectos menores, en algunas capacitaciones y prácticas, pero aún no se integra en los procesos de despliegue oficiales.
3	Uso de Git u otro VCS (control de versiones) de forma centralizada y con buenas prácticas	<b>Medio</b>	Se utiliza Git como repositorio principal, pero no existen normas formales ni estandarizadas para la estructura de ramas, commits o revisiones de código.
4	Políticas formales documentadas para ramas, merge, revisiones de código	<b>Bajo</b>	No hay lineamientos documentados; cada equipo o desarrollador aplica su propio criterio y esto genera inconsistencias.
5	Pipeline automatizado de compilación / build	<b>Bajo</b>	Existen pipelines para la automatización en la compilación, pero dependen del trabajo humano, ya que con frecuencia muestran errores.
6	Pipeline automatizado de pruebas (unitarias / integración)	<b>Bajo</b>	No hay pruebas automatizadas. Las validaciones dependen del equipo de QA y se hacen manualmente.
7	Pipeline automatizado de despliegue (despliegue continuo / entrega continua)	<b>Bajo</b>	Existen pipelines para la automatización en los despliegues, pero dependen del trabajo humano, ya que con frecuencia muestran errores.
8	Políticas de rollback / recuperación ante fallo documentadas	<b>Medio</b>	Existen procedimientos básicos de respaldo y recuperación, pero no están documentados formalmente ni automatizados.
9	Integración de monitoreo / logging dentro del pipeline	<b>Medio</b>	Existe una configuración de monitoreo básico.
10	Políticas de versionado	<b>Medio</b>	El versionado se realiza, pero de forma inconsistente; no hay una política estándar para numeración o manejo de releases.
11	Políticas definidas para automatización de entornos (de prueba, staging, producción)	<b>Medio</b>	Cuenta con entornos separados, pero la creación y configuración aún dependen del trabajo manual.
12	Prácticas de revisión continua (code review, pull requests) integradas al flujo	<b>Medio</b>	Se realizan revisiones de código, pero no en todos los proyectos ni con criterios homogéneos.

13	Políticas estandarizadas para manejo de entornos (infraestructura como código, configuración automática)	<b>Medio</b>	La estandarización de entornos depende de procesos manuales que generan diferencias entre ambientes.
----	--	--------------	--

Nota: Elaboración propia.

Los resultados muestran que, de los trece ítems evaluados, la mayoría se encuentran en niveles de capacidad bajos o medios, sin identificarse ningún elemento con nivel alto:

- Ninguno de los trece elementos evaluados alcanzó un nivel alto; predominaron niveles bajo y medio, lo que evidencia una madurez limitada para adoptar prácticas DevOps.
- Existen brechas severas en procesos, destacando la ausencia total de pipelines automatizados de compilación, pruebas y despliegues, así como la falta de políticas formales para ramas, merges y revisiones de código.
- En cuanto a herramientas, Git y Docker se utilizan parcialmente y sin estandarización, mientras que las herramientas de monitoreo no están integradas al ciclo de desarrollo ni al pipeline.
- En competencias, no se cuenta con prácticas maduras de automatización de entornos ni infraestructura como código; además, las revisiones de código se aplican de manera inconsistente.
- En conjunto, los resultados muestran brechas estructurales que limitan la implementación de un modelo CI/CD robusto, señalando la necesidad urgente de estandarización, automatización y fortalecimiento de competencias técnicas alineadas con DevOps.

Los resultados del Checklist técnico revelan brechas estructurales en NGS que afectan directamente la capacidad de implementar un modelo CI/CD robusto: falta de automatización, procesos manuales, ausencia de políticas y estandarización, uso parcial de herramientas y competencias técnicas aún en desarrollo. Estas brechas confirman la necesidad de fortalecer procesos, formalizar prácticas y avanzar hacia una mayor madurez tecnológica alineada con DevOps.

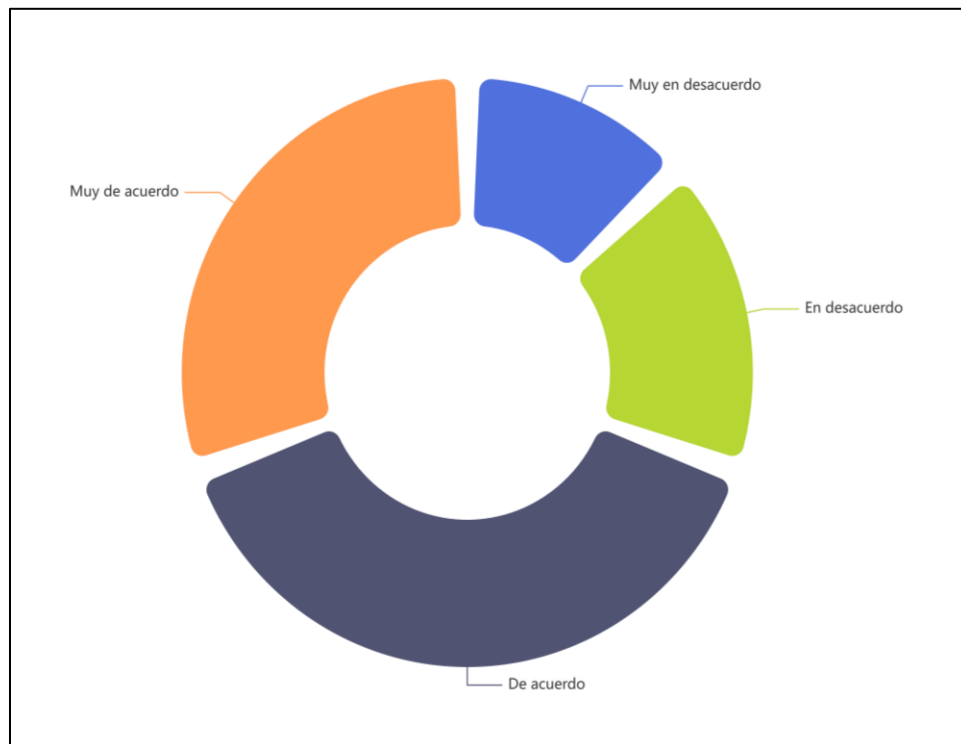
## ENCUESTA: PROCESOS Y BRECHAS ACTUALES

Posterior a la aplicación del Checklist técnico, el cual permitió identificar brechas operativas relacionadas con la automatización, el uso de herramientas y la estandarización de prácticas, se procedió a aplicar la Sección A de la encuesta con el propósito de complementar este diagnóstico desde la perspectiva del personal técnico de NGS. Esta sección se centró en evaluar cómo los colaboradores perciben los procesos actuales de desarrollo y despliegue, profundizando en aspectos como documentación, flujos de trabajo, claridad de roles, comunicación entre equipos y nivel de automatización. La información recabada permite contrastar la visión directiva obtenida en el Checklist con la experiencia diaria del equipo, ofreciendo una comprensión más completa de las debilidades existentes en los procesos y evidenciando las áreas donde las limitaciones técnicas y organizacionales impactan directamente en la eficiencia del ciclo de vida del software.

A continuación, se presentan y describen las preguntas y respuestas desarrolladas:

1. Los procesos actuales de desarrollo y despliegue están documentados y son de conocimiento general.

**Figura 30** Documentación de procesos de desarrollo y despliegue



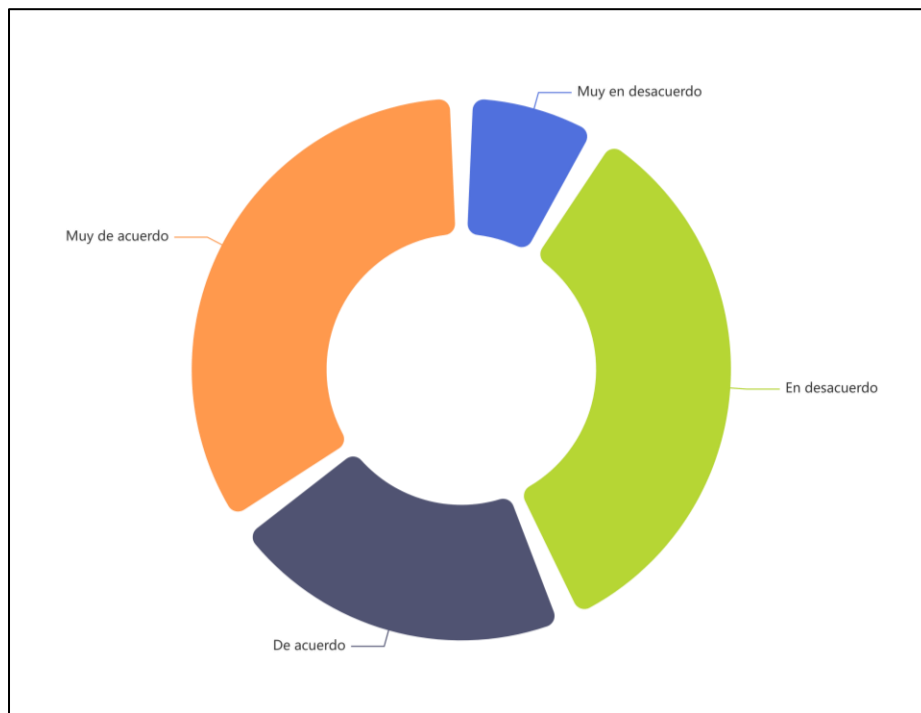
Nota: Elaboración propia.

Los participantes muestran una percepción predominantemente positiva respecto a la existencia y conocimiento de los procesos actuales de desarrollo y despliegue. La mayoría coincide en que la documentación existe y es conocida dentro del equipo, pero no de manera plenamente formalizada. Sin embargo, un grupo significativo señala que dicha documentación es limitada o insuficiente, lo que evidencia que aún persisten vacíos considerables.

Esto indica que la organización cuenta con una base documental funcional, pero incompleta. La información no está estructurada bajo un estándar institucional, y su calidad depende en gran medida del conocimiento individual de los colaboradores. Esta condición provoca fragilidad en la transferencia interna de conocimientos y afecta la consistencia del ciclo de desarrollo y despliegue. Además, se relaciona directamente con una de las debilidades identificadas en el FODA: la carencia de documentación oficial, clara y homogénea. En consecuencia, la trazabilidad y la calidad del flujo CI/CD se ven parcialmente comprometidas.

2. Los flujos de trabajo de desarrollo y entrega se encuentran estandarizados y definidos formalmente.

**Figura 31** Estandarización de los flujos de trabajo de desarrollo y entrega



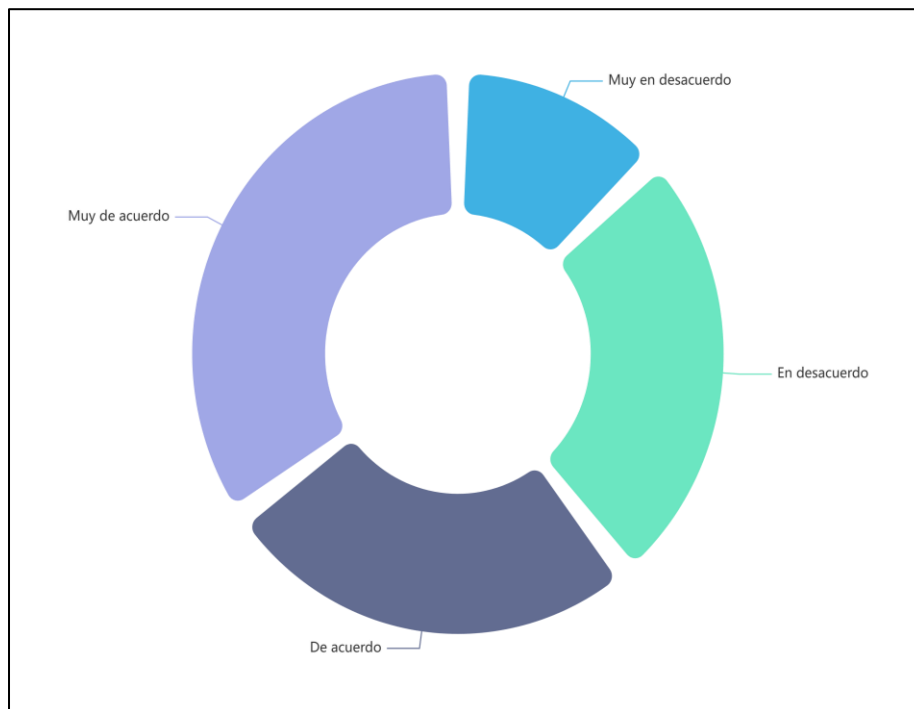
Nota: Elaboración propia.

La percepción del personal evidencia que una parte considerable del equipo está de acuerdo o muy de acuerdo en que los flujos de trabajo cuentan con cierto nivel de estandarización y funcionan de manera aceptable. Esto muestra que, para muchos colaboradores, existen lineamientos básicos que permiten mantener un orden en las actividades del ciclo de desarrollo y despliegue.

No obstante, también se identifica un grupo relevante que considera que aún hay informalidad o falta de consistencia en los flujos. Esta percepción indica que los procedimientos no se aplican de forma uniforme en toda la organización: algunos equipos siguen prácticas más estructuradas, mientras que otros operan con mayor variabilidad o sin un estándar claro. El resultado cualitativo refleja que la estandarización actual es parcial. Aunque existe una base que genera acuerdos y buenas prácticas, la organización todavía enfrenta diferencias notables entre equipos y proyectos, lo que provoca inconsistencias, retrasos y limita el avance hacia niveles más altos de automatización en CI/CD.

### 3. Existen roles y responsabilidades claramente definidos en el proceso CI/CD.

**Figura 32** Claridad en roles y responsabilidades dentro del proceso CI/CD



Nota: Elaboración propia.

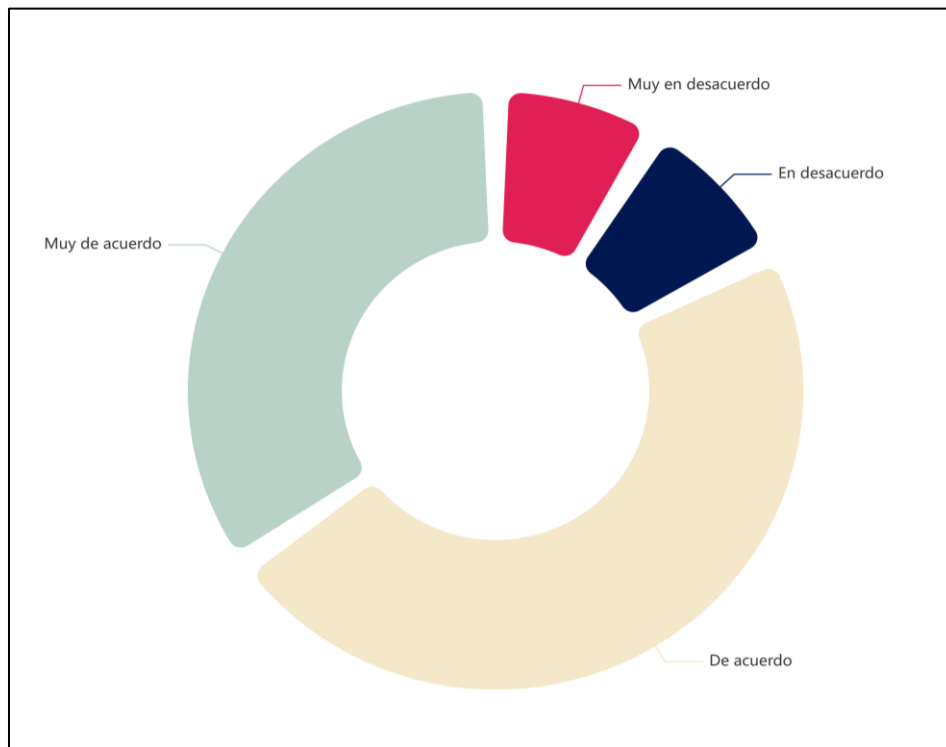
Los resultados muestran que una parte importante del personal está de acuerdo o muy de

acuerdo en que existen roles y responsabilidades con cierto nivel de claridad dentro del proceso CI/CD. Para este grupo, la organización ha logrado establecer una base mínima de definición que les permite comprender sus funciones y cómo encajan dentro del flujo de trabajo. No obstante, también se identifica un porcentaje significativo de colaboradores que están en desacuerdo o muy en desacuerdo respecto a la claridad de los roles. Estos participantes perciben que la definición actual es insuficiente, ambigua o poco formalizada, lo que genera dudas sobre quién debe ejecutar determinadas tareas o tomar decisiones en momentos críticos.

El hallazgo coincide con lo identificado en el FODA, donde la ausencia de roles formalmente establecidos se reconoció como una debilidad estructural que limita la eficiencia del proceso de entrega continua.

4. Las herramientas utilizadas son adecuadas para las necesidades del equipo.

**Figura 33** *Adecuación de las herramientas utilizadas por el equipo*



Nota: Elaboración propia.

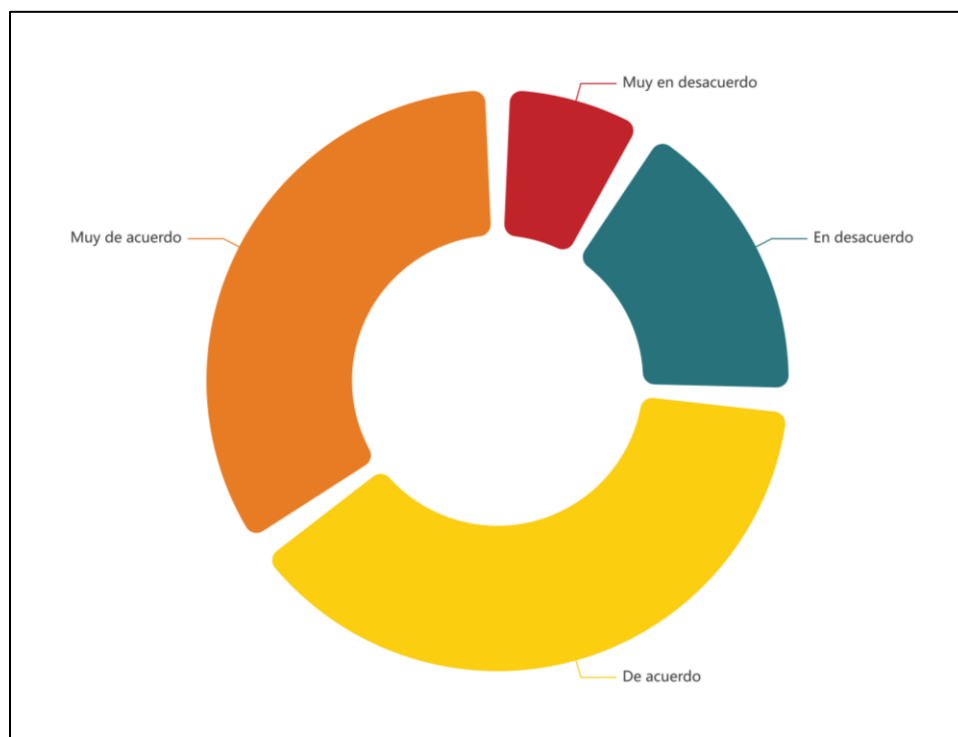
La mayoría del equipo expresa que las herramientas utilizadas son adecuadas para las necesidades actuales, ubicándose en niveles de acuerdo y muy de acuerdo. Esto refleja que la organización cuenta con un conjunto de herramientas modernas y funcionales que permiten

desarrollar y gestionar las actividades técnicas de manera eficiente.

Los datos muestran una fortaleza clara: NGS cuenta con herramientas modernas (por ejemplo, Azure DevOps, repositorios y sistemas de versionamiento). Sin embargo, la percepción minoritaria de insuficiencia sugiere posibles brechas en uso avanzado, configuración o capacitación, más que limitaciones de la herramienta en sí.

5. El personal técnico posee las competencias necesarias para ejecutar prácticas de CI/CD y automatización.

**Figura 34** Competencias del personal técnico para prácticas CI/CD y automatización



Nota: Elaboración propia.

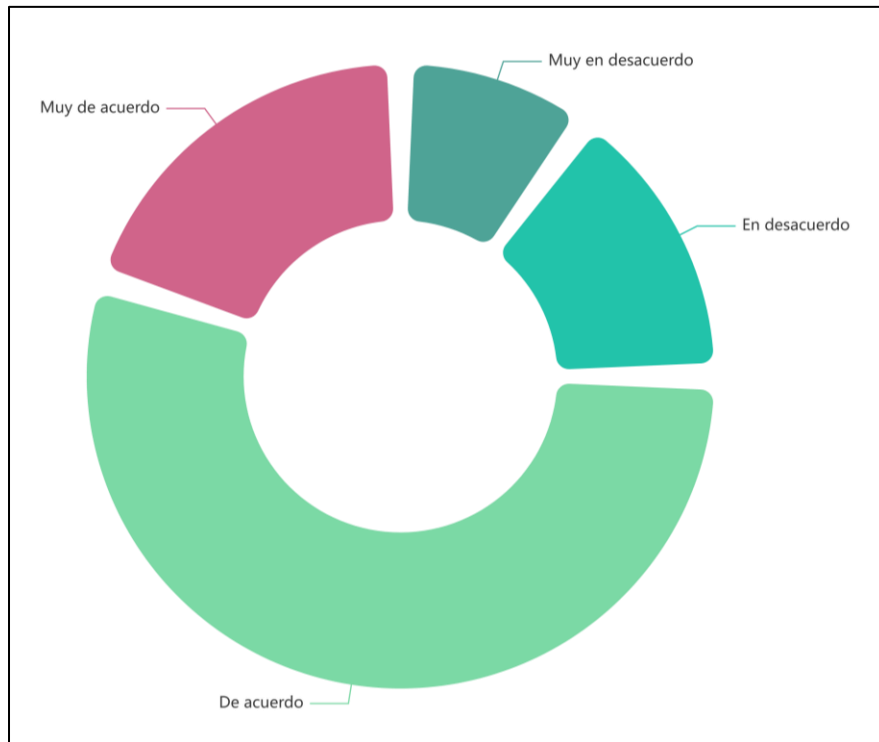
La mayoría del equipo percibe que posee competencias técnicas adecuadas para trabajar con prácticas CI/CD y procesos de automatización. Esta valoración, situada entre acuerdo y muy de acuerdo, indica que existe una base de conocimientos sólida y funcional dentro del personal técnico.

Los resultados muestran que existe una base técnica sólida en el equipo, pero aún no homogénea. Las brechas señaladas reflejan diferencias de experiencia entre equipos y apuntan a la necesidad de fortalecer competencias específicas en automatización, pipelines y pruebas

automáticas.

6. El flujo de entrega continua actual es eficiente y permite entregas frecuentes y controladas.

**Figura 35** Eficiencia del flujo actual de entrega continua



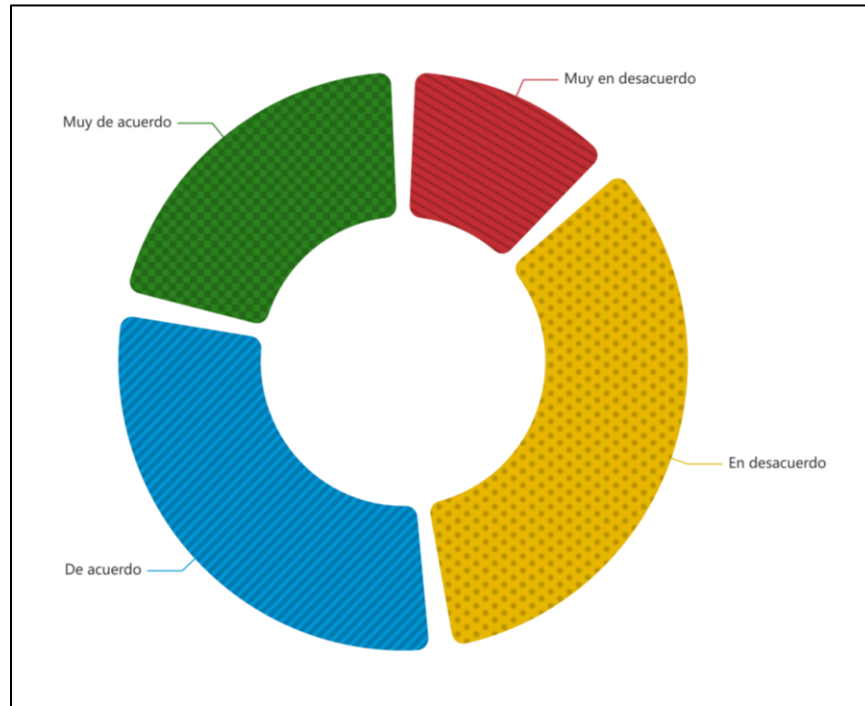
Nota: Elaboración propia.

La mayoría del equipo percibe que el flujo de entrega continua funciona de manera aceptable y muestra un nivel razonable de eficiencia. Quienes se ubican en acuerdo o muy de acuerdo consideran que el proceso permite avanzar y cumplir con las entregas, aunque reconociendo que aún existen oportunidades de mejora.

Aunque el flujo opera de manera aceptable, aún no alcanza niveles óptimos. Factores como procesos manuales, roles poco definidos y documentación insuficiente inciden directamente en esta percepción, evidenciando que la eficiencia actual depende más del esfuerzo del equipo que de procesos estandarizados.

7. Existen retrasos frecuentes por actividades manuales en pruebas o despliegues.

**Figura 36** Retrasos por actividades manuales en pruebas y despliegues



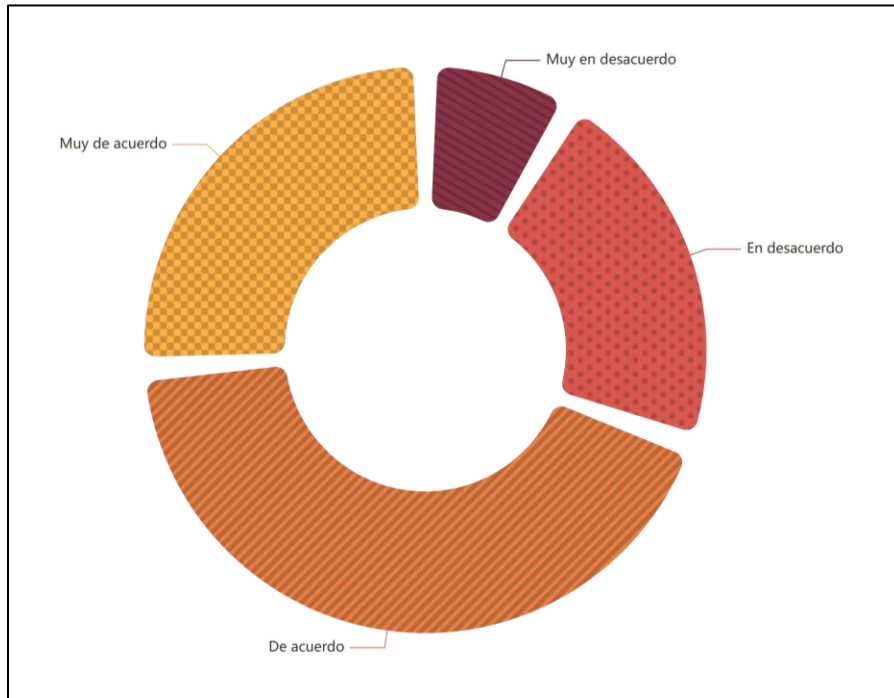
Nota: Elaboración propia.

Una parte importante del equipo reconoce que los retrasos en el flujo de trabajo se deben a la presencia de tareas manuales en pruebas y despliegues. Quienes se ubican en acuerdo o muy de acuerdo coinciden en que estas actividades manuales generan demoras perceptibles y afectan la continuidad del proceso.

Este hallazgo apoya una de las principales brechas técnicas: la dependencia de actividades manuales en pruebas y despliegues, lo cual afecta tiempos, añade riesgo operativo y limita la escalabilidad del flujo CI/CD. Este punto coincide con varias respuestas abiertas que sugieren priorizar la automatización.

8. Existen mecanismos automatizados para el despliegue de nuevas versiones en los entornos de prueba o producción.

**Figura 37** Existencia de mecanismos automatizados de despliegue



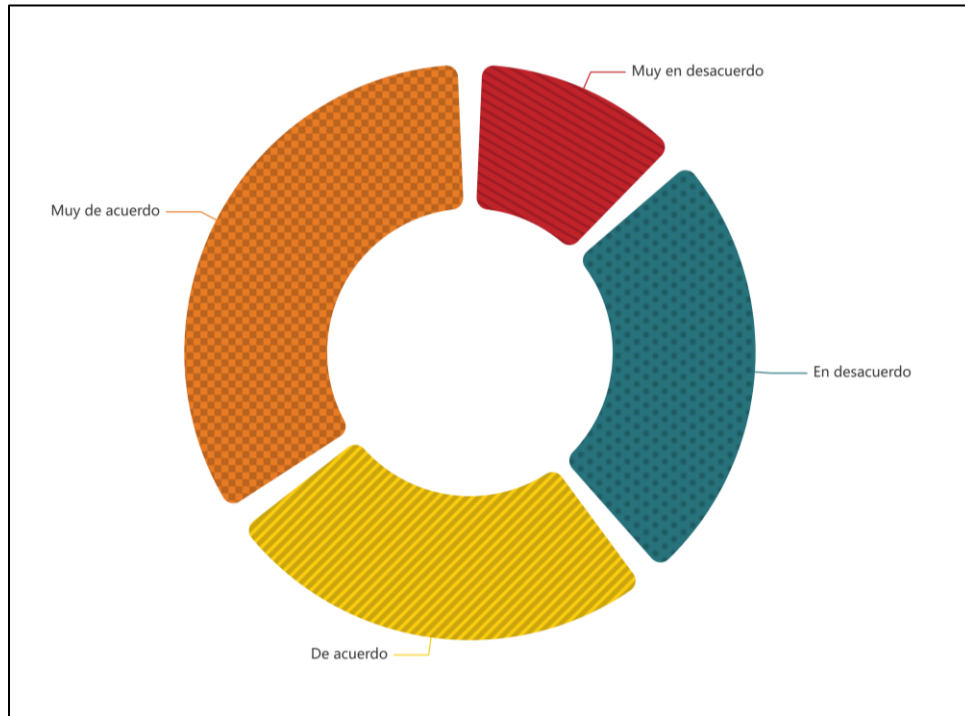
Nota: Elaboración propia.

La mayoría del equipo percibe que existen mecanismos de automatización para los despliegues, ubicándose en acuerdo o muy de acuerdo. Esto indica que la automatización está presente y forma parte del proceso, al menos en ciertos niveles o para determinados proyectos.

La automatización está presente, pero no completamente integrada ni estandarizada. Esto indica una madurez intermedia donde la automatización depende más del equipo que del proceso institucional. Esto representa una oportunidad clara para fortalecer CI/CD con pipelines homogéneos y reproducibles.

9. Las herramientas de integración y despliegue continuo (por ejemplo, Jenkins, GitLab CI, Azure DevOps) están correctamente configuradas e integradas entre sí.

**Figura 38** Configuración e integración de herramientas CI/CD



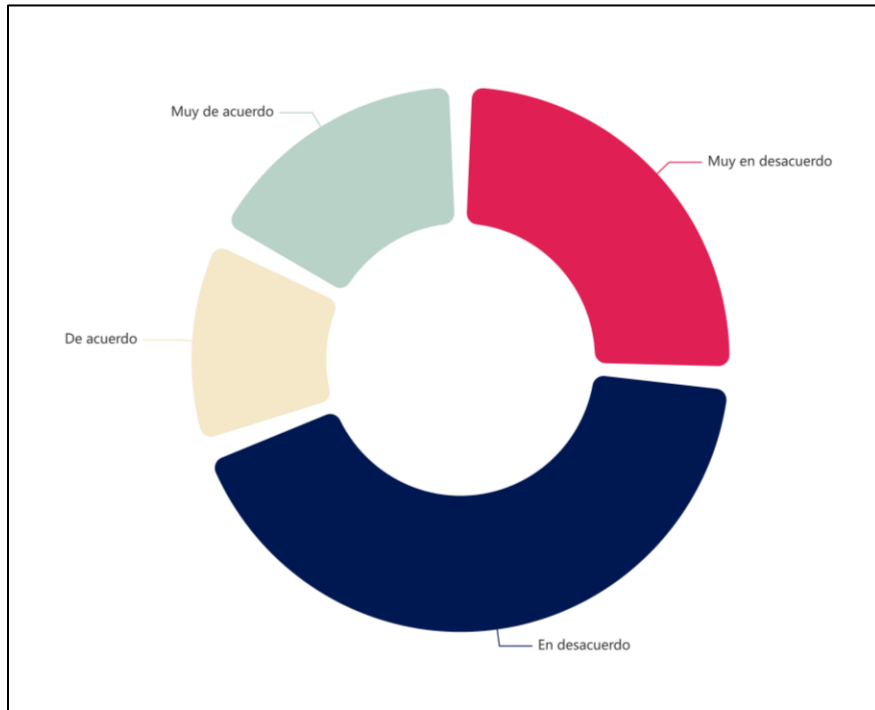
Nota: Elaboración propia.

La mayoría del equipo percibe que las herramientas utilizadas para CI/CD están configuradas de manera adecuada. Esto indica que, en general, el conjunto de herramientas funciona y se encuentra operativamente alineado con las necesidades básicas del proceso.

Aunque existe una tendencia positiva, las respuestas sugieren que las integraciones aún no son uniformes en todos los proyectos. Esto limita la confiabilidad y consistencia del proceso CI/CD, ya que configuraciones dispares generan resultados distintos para equipos distintos.

10. Con frecuencia se capacita el personal técnico para el uso de nuevas herramientas o prácticas de automatización.

**Figura 39** Frecuencia de capacitación en nuevas herramientas o prácticas de automatización



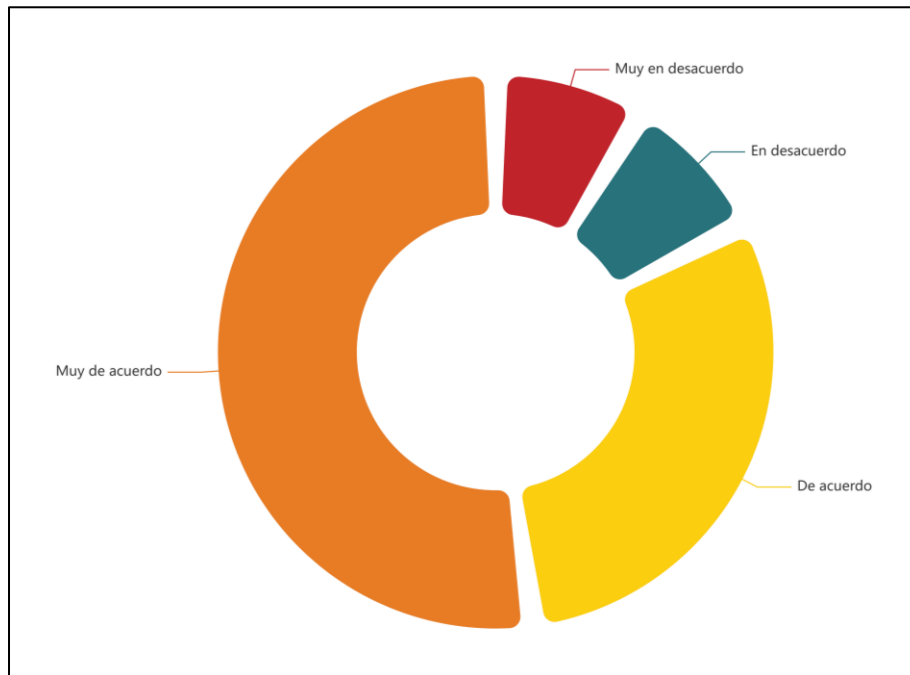
Nota: Elaboración propia.

Un porcentaje alto del equipo considera que la capacitación en nuevas herramientas o prácticas de automatización es poco frecuente, coinciden en que las oportunidades de formación continua no son suficientes para sostener la actualización técnica que exige el proceso CI/CD.

Este es uno de los hallazgos más críticos: la falta de capacitación continua afecta directamente las competencias del personal y la velocidad de adopción de nuevas prácticas. Aunque NGS ofrece programas formativos, los resultados indican que no se están aprovechando o no están alineados con las necesidades reales del equipo.

11. Considera que la falta de documentación o estandarización afecta la calidad y velocidad de las entregas.

**Figura 40** Impacto de la falta de documentación y estandarización en la calidad y velocidad de las entregas



Nota: Elaboración propia.

La mayoría del equipo percibe claramente que la falta de documentación y estandarización afecta de manera directa la calidad y la velocidad de las entregas, reconocen que esta ausencia genera retrabajos, inconsistencias y retrasos en el proceso, influyendo negativamente en el desempeño global del equipo.

Los participantes tienen una percepción clara sobre cómo la ausencia de documentación afecta la calidad, genera retrabajos y ralentiza las entregas. Este hallazgo refuerza la necesidad de crear lineamientos institucionales para garantizar consistencia y eficiencia.

12. Desde su experiencia, ¿Qué acciones considera más importantes para mejorar los procesos, herramientas o habilidades del equipo que interviene en el desarrollo, pruebas y despliegue de software?

**Tabla 22** Descripción y análisis de las respuestas abiertas

Categoría identificada	Descripción general de la necesidad expresada	Ejemplos representativos de los participantes	Análisis
<p><b>1. Capacitación continua y desarrollo de habilidades técnicas</b></p>	<p>Los participantes destacan la necesidad de mayor formación en herramientas, CI/CD, automatización y buenas prácticas.</p>	<p>Capacitación continua para el uso de las herramientas implementadas. Capacitaciones periódicas sobre CI/CD, nuevas herramientas y tendencias. Es indispensable mantener informado y capacitado a todo el recurso humano.</p>	<p>La capacitación es la necesidad más reiterada. Existe percepción de brechas en conocimientos para automatización, despliegue y uso avanzado de herramientas. Coincide con los hallazgos del FODA sobre bajo aprovechamiento de programas formativos.</p>
<p><b>2. Estandarización y documentación de procesos</b></p>	<p>Solicitan procesos formales, homogéneos y documentados entre equipos.</p>	<p>Es importante estandarizar los procesos entre todos los equipos de desarrollo. Tener una correcta documentación de los procesos. Estandarizar los procesos de manera oficial.</p>	<p>La ausencia de estándares unificados genera variabilidad y retrabajos. Esta carencia afecta CI/CD, calidad, tiempos y consistencia. Es una brecha crítica señalada también por los resultados de las preguntas 1, 2 y 11 de la sección A.</p>
<p><b>3. Mayor automatización del ciclo de desarrollo, pruebas y despliegue</b></p>	<p>Indican la necesidad de automatizar entregas, pruebas e integraciones para reducir errores y tiempos.</p>	<p>Fortalecer la parte de automatización. Usar herramientas que den consistencia, automatización y buena observabilidad. Trabajar en automatización de pruebas y despliegue.</p>	<p>La automatización se reconoce como pilar para mejorar la eficiencia y reducir tareas manuales. Actualmente, el nivel de automatización es desigual entre equipos y depende del conocimiento individual.</p>
<p><b>4. Comunicación, colaboración y cultura organizacional</b></p>	<p>Señalan la necesidad de mejorar la interacción entre equipos Dev, QA y Ops, así como la gestión del tiempo y la cooperación.</p>	<p>Comunicación y gestión del tiempo. Fomentar una cultura de colaboración entre Dev, QA y Ops. Robustecer la cooperación entre operaciones, desarrollo y pruebas.</p>	<p>Existen “silos” entre equipos que afectan la coordinación, el flujo de entrega y la calidad. La madurez DevOps requiere mayor alineación cultural, no solo técnica.</p>
<p><b>5. Mejor uso e implementación de herramientas</b></p>	<p>Proponen optimizar la integración y uso de herramientas existentes como Azure DevOps, repositorios, automatización, etc.</p>	<p>Mejorar la integración con herramientas CI/CD. Asegurar que las herramientas se adapten a las necesidades del negocio.</p>	<p>Aunque las herramientas actuales son consideradas adecuadas, su uso no siempre es uniforme ni configurado bajo un estándar, lo cual afecta la consistencia del pipeline.</p>

<p><b>6. Fortalecimiento del talento humano y reducción de variabilidad</b></p>	<p>Se menciona la importancia de valorar y potenciar las capacidades del personal técnico.</p>	<p>Fortalecer la capacidad de programadores y analistas en lógica y buenas prácticas. Disponer de herramientas que faciliten las fases del desarrollo del producto.</p>	<p>Refuerza que NGS cuenta con personal altamente capacitado, pero requiere lineamientos formales que permitan aprovechar su potencial de forma sostenible.</p>
---	--	---	---

Nota: Elaboración propia basada en la pregunta 12 de la sección A de la encuesta.

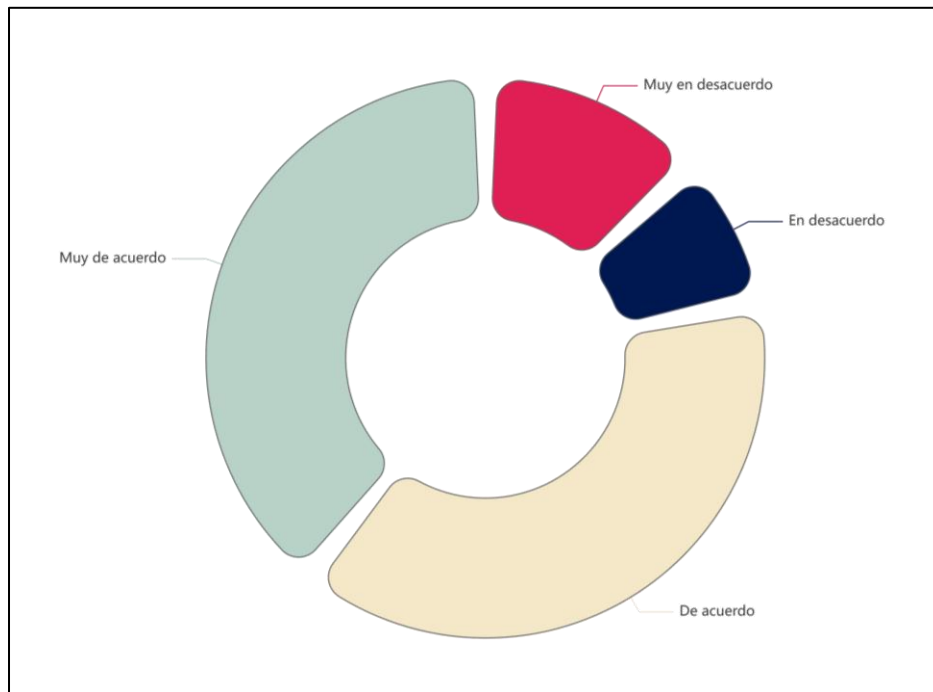
**ENCUESTA: INTEGRACIÓN DEVOPS**

La encuesta de la Integración DevOps busca conocer hasta qué punto las herramientas y prácticas DevOps, como Jenkins, Docker o Git, están presentes en el trabajo diario de NGS y si existen políticas que apoyen su uso de manera consistente. A través de una escala Likert de cuatro puntos, los participantes evaluaron afirmaciones relacionadas con automatización, integración continua y colaboración, permitiendo identificar el nivel real de adopción de estas prácticas dentro de la organización.

A continuación, se muestran las preguntas y respuestas relacionadas con esta sección:

1. Se aplican prácticas DevOps en las etapas de desarrollo, pruebas y despliegue.

**Figura 41** *Aplicación de prácticas DevOps en desarrollo, pruebas y despliegue*



Nota: Elaboración propia.

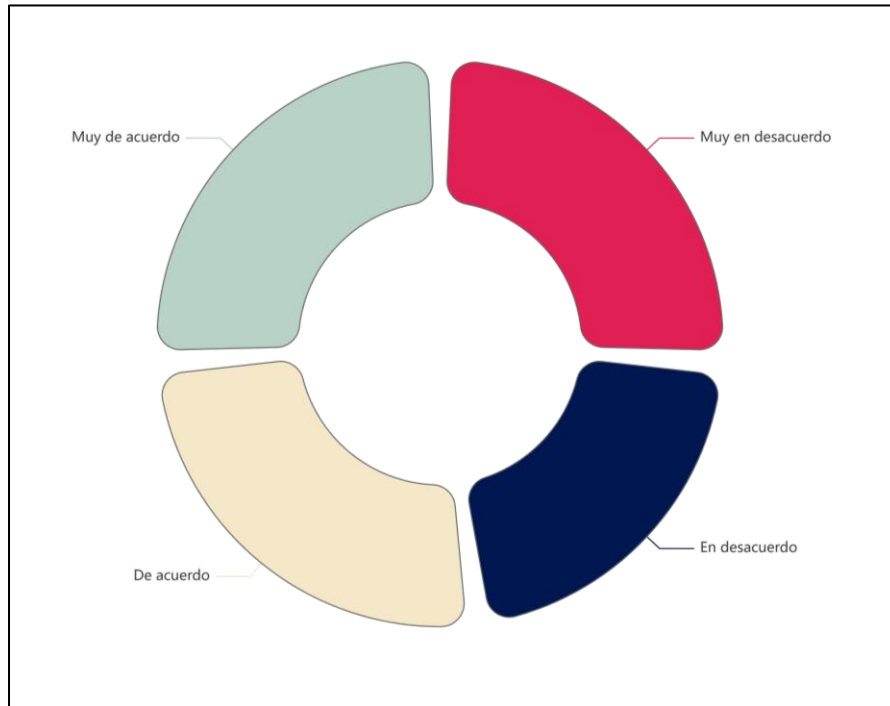
Los resultados muestran que la mayoría de los participantes percibe que las prácticas

DevOps sí se aplican en las etapas de desarrollo, pruebas y despliegue dentro de NGS. Las respuestas “De acuerdo” y “Muy de acuerdo” representan el porcentaje más alto, indicando que existe una adopción significativa de prácticas relacionadas con automatización, colaboración entre equipos y mejoras en el flujo de entrega.

Podemos ver que una proporción menor de encuestados seleccionó “En desacuerdo” y “Muy en desacuerdo”, lo que evidencia que, si bien las prácticas DevOps están presentes, su aplicación no es homogénea en todos los proyectos o equipos. Esto sugiere que aún existen áreas donde estas prácticas no se encuentran completamente integradas o se aplican de manera parcial. Los resultados evidencian que NGS ha avanzado en la adopción de prácticas DevOps dentro del ciclo de desarrollo, pruebas y despliegue, lo cual constituye una base importante para fortalecer el flujo CI/CD. Sin embargo, la presencia de percepciones divergentes muestra que la implementación no es totalmente uniforme, lo que indica la necesidad de continuar estandarizando procesos, mejorar la integración entre equipos y ampliar la automatización en puntos críticos. En conjunto, el gráfico confirma un nivel de madurez intermedio en DevOps: existen prácticas bien encaminadas, pero aún requieren mayor formalización y consistencia para consolidarse como un modelo plenamente integrado en toda la organización.

2. Se llevan a cabo pruebas automatizadas como parte del proceso de integración continua.

**Figura 42** Nivel de adopción de pruebas automatizadas en el proceso de integración continua



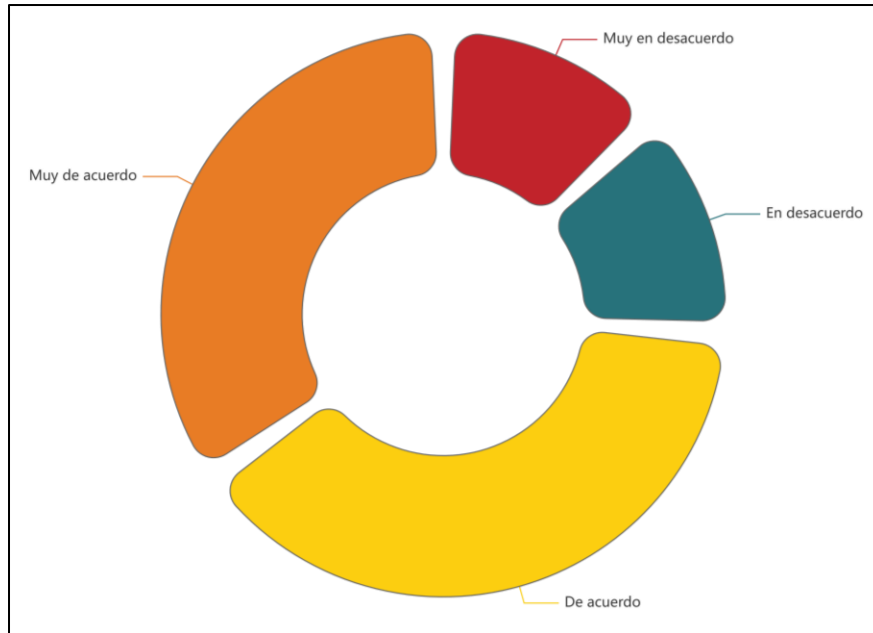
Nota: Elaboración propia.

El gráfico evidencia una percepción distribuida respecto a la adopción de pruebas automatizadas en el proceso de integración continua. Aunque existe un grupo de participantes que seleccionó “De acuerdo” y “Muy de acuerdo”, lo que refleja avances en la implementación de pruebas automatizadas, también se observa una proporción significativa de respuestas en “En desacuerdo” y “Muy en desacuerdo”. Esto indica que la práctica de automatizar pruebas está presente en algunos proyectos o equipos, pero aún no se encuentra estandarizada ni aplicada de manera uniforme en toda la organización. Las diferencias en las respuestas sugieren que la automatización depende del tipo de proyecto, del equipo involucrado o del nivel de madurez de cada flujo de desarrollo. Los resultados nos permiten concluir que la automatización de pruebas en NGS se encuentra en un nivel de adopción intermedio. Aunque existen iniciativas y esfuerzos por integrar pruebas automatizadas en el flujo CI/CD, la presencia de respuestas negativas demuestra que esta práctica aún no está consolidada ni implementada de forma sistemática. Para avanzar hacia un modelo DevOps más robusto, será necesario fortalecer la estandarización, ampliar la cobertura de pruebas automatizadas y garantizar su ejecución continua en todos los proyectos. Esto permitirá mejorar la calidad del software, reducir errores en despliegues y acelerar

los ciclos de entrega continua.

3. Existe una cultura de colaboración activa entre los equipos de desarrollo, QA y operaciones.

**Figura 43** *Colaboración entre equipos de desarrollo, QA y operaciones*



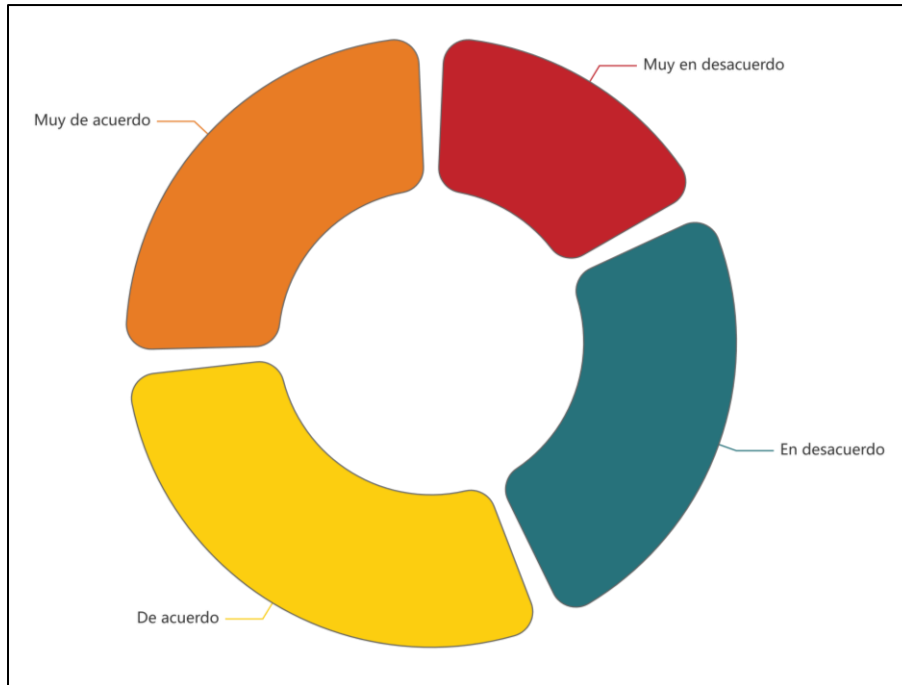
Nota: Elaboración propia.

Los resultados muestran que la mayoría de los participantes percibe que sí existe una cultura de colaboración entre los equipos de Desarrollo, QA y Operaciones. Las respuestas “De acuerdo” y “Muy de acuerdo” representan el porcentaje más alto, lo que evidencia que la interacción y el trabajo conjunto entre las áreas es un aspecto visible dentro de NGS. Esto sugiere que los equipos mantienen una comunicación activa y coordinan esfuerzos para avanzar en los procesos de desarrollo, validación y despliegue, también se observan respuestas en “En desacuerdo” y “Muy en desacuerdo”, aunque en menor proporción. Esto indica que la colaboración no es completamente homogénea y que existen proyectos o equipos donde esta interacción podría ser limitada o no tan fluida, reflejando variaciones en prácticas o en dinámicas de trabajo. NGS presenta una base positiva en cuanto a la colaboración entre Desarrollo, QA y Operaciones, lo cual es un pilar fundamental para la adopción de DevOps. Sin embargo, las percepciones divergentes muestran que esta colaboración aún no se encuentra plenamente consolidada en todas las áreas. Para fortalecer la madurez DevOps, será necesario promover prácticas colaborativas más

estandarizadas, mejorar los canales de comunicación y fomentar una cultura de trabajo integrada que facilite la entrega continua y la mejora del flujo CI/CD.

4. Disponemos de políticas formales para ramas, merge y revisión de código.

**Figura 44** Disponibilidad de políticas formales para ramas, merge y revisión de código



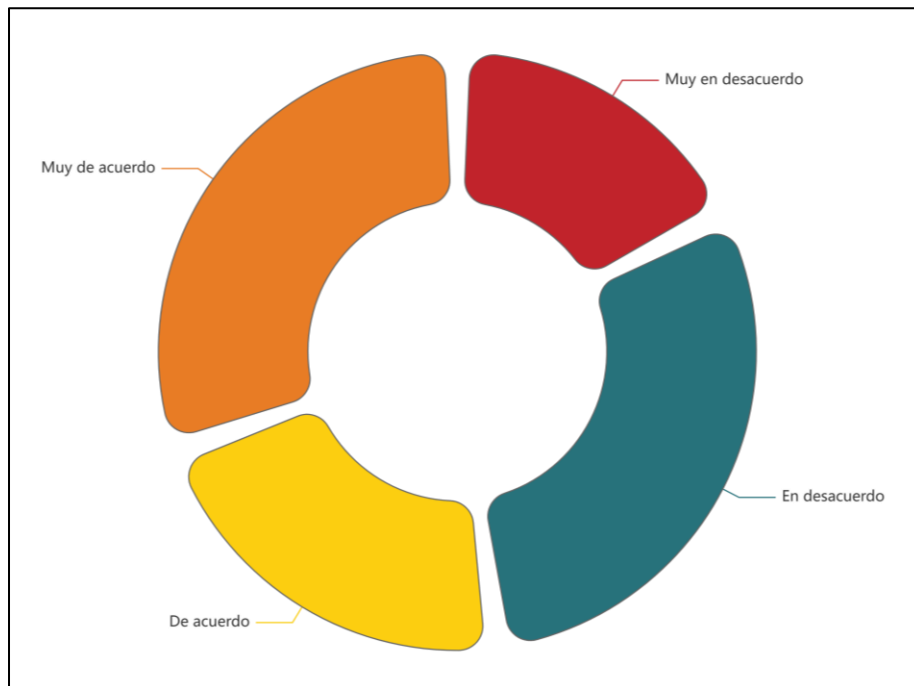
Nota: Elaboración propia.

Los resultados nos muestran una distribución equilibrada en las percepciones sobre la existencia de políticas formales para la gestión de ramas, procesos de merge y revisión de código en NGS. Una parte importante de los participantes seleccionó “De acuerdo” y “Muy de acuerdo”, lo que indica que en ciertos equipos o proyectos sí existen lineamientos definidos para controlar la calidad del código y la integración entre desarrolladores. Esto sugiere cierto nivel de madurez en prácticas esenciales para prevenir errores y asegurar estabilidad en los despliegues, también se observa un porcentaje significativo de respuestas en “En desacuerdo” y “Muy en desacuerdo”, lo que evidencia que estas políticas no están completamente estandarizadas ni aplicadas de manera uniforme. Esto puede reflejar diferencias entre proyectos, falta de documentación o ausencia de un marco común que regule el versionamiento y la colaboración en el código. La presencia de percepciones negativas demuestra que estas prácticas no están formalizadas de manera consistente en todos los equipos. Para fortalecer la madurez DevOps, NGS debe consolidar lineamientos

unificados, asegurar su cumplimiento y promover la revisión sistemática del código como parte del ciclo de entrega. Esto contribuirá a reducir errores, mejorar la trazabilidad y garantizar integraciones más seguras y controladas.

5. Incorporamos monitoreo y logs dentro del pipeline con alertas.

**Figura 45** *Uso de monitoreo y gestión de logs integrados en el pipeline con alertas*

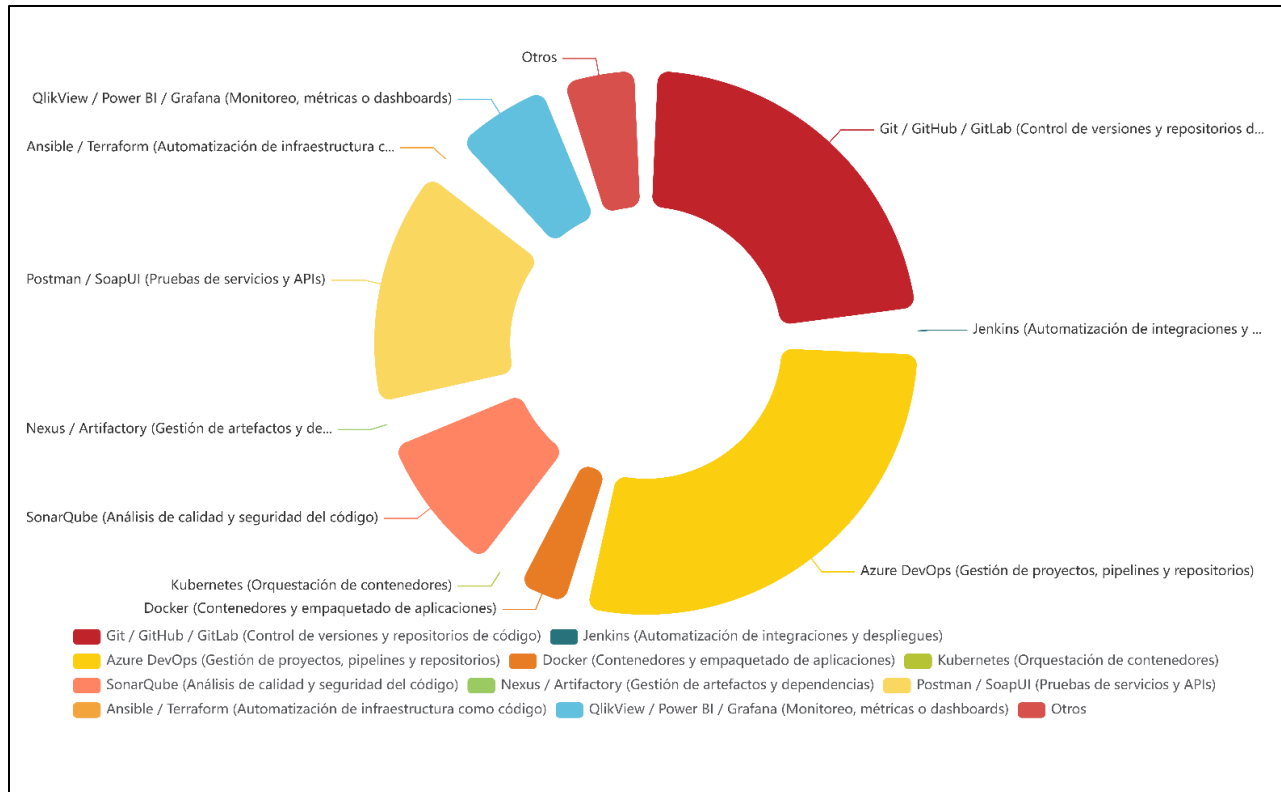


Nota: Elaboración propia.

El gráfico muestra opiniones divididas sobre el uso de monitoreo y logs integrados en el pipeline con alertas. Aunque una parte de los encuestados está de acuerdo o muy de acuerdo, lo que indica avances en la observabilidad dentro de los pipelines, también existe un grupo significativo en desacuerdo, lo que evidencia que estas prácticas no se aplican de forma consistente en todos los equipos. NGS ha comenzado a integrar monitoreo y gestión de logs con alertas en sus pipelines, pero su implementación sigue siendo desigual. Para fortalecer su madurez DevOps, es necesario estandarizar herramientas y asegurar que todos los proyectos cuenten con monitoreo proactivo y mecanismos de alerta que permitan detectar fallos oportunamente.

6. ¿Cuáles de las siguientes herramientas o plataformas utiliza actualmente en su área o equipo dentro del proceso de desarrollo, integración o despliegue (CI/CD)?

**Figura 46 Herramientas y plataformas utilizadas actualmente en los equipos para CI/CD**



Nota: Elaboración propia.

El gráfico muestra una amplia variedad de herramientas utilizadas por los equipos de NGS dentro del proceso de desarrollo, integración y despliegue. Las plataformas con mayor presencia son Git/GitHub/GitLab, asociadas al control de versiones; Azure DevOps, utilizado para la gestión de proyectos y repositorios; y Jenkins, herramienta clave para la automatización de integraciones y despliegues. También se identifican herramientas orientadas a pruebas (Postman/SOAPUI), calidad de código (SonarQube), contenedores (Docker/Kubernetes) y gestión de artefactos (Nexus/Artifactory). En menor proporción aparecen soluciones para automatización de infraestructura (Ansible/Terraform) y monitoreo (Grafana/Power BI), junto con menciones a “Otros”. De los resultados podemos decir que NGS utiliza una combinación diversa de herramientas alineadas con prácticas modernas de CI/CD y DevOps. Sin embargo, la variedad de soluciones y su distribución desigual sugieren que no existe una estandarización completa entre equipos. Avanzar hacia un conjunto de herramientas unificadas y configuradas de manera consistente permitirá mejorar la integración, la automatización y la trazabilidad en todo el ciclo de desarrollo y operación.

7. Desde su experiencia, ¿Qué prácticas o herramientas DevOps considera más necesarias para mejorar el flujo de trabajo en NGS?

**Tabla 23** Descripción y análisis de las respuestas abiertas

Aspecto clave	Descripción breve	Análisis resumido
<b>Control de versiones (Git)</b>	Uso de GitHub/GitLab para manejo ordenado del código.	Garantiza trazabilidad y trabajo colaborativo seguro.
<b>Pipelines CI/CD</b>	Despliegues y construcciones automáticas.	Reduce errores manuales y acelera entregas.
<b>Pruebas automatizadas</b>	Unitarias, integración y regresión.	Mejora la calidad y evita fallas en producción.
<b>Calidad del código (SonarQube)</b>	Análisis estático de código.	Disminuye deuda técnica y aumenta confiabilidad.
<b>Metodologías ágiles</b>	Implementación de Scrum/Kanban.	Mejora la organización y el flujo de trabajo DevOps.
<b>Capacitación y documentación</b>	Formación continua y minutas claras.	Fortalece habilidades y evita malentendidos.
<b>Replicación de ambientes (Docker)</b>	Ambientes locales similares a los del cliente.	Facilita pruebas consistentes y reduce dependencia.
<b>Coherencia entre equipos y proyectos</b>	Adoptar buenas prácticas comunes.	Alinea procesos y fortalece la cultura DevOps.

Nota: Elaboración propia basada en la pregunta 7 de la sección B de la encuesta.

Luego de la aplicación de los instrumentos correspondientes al primer objetivo, procedemos a profundizar en los resultados, análisis y sus principales hallazgos correspondiente a las tres dimensiones evaluadas: procesos, herramientas y competencias. A continuación se detallan estas dimensiones:

- a) **Brechas en los procesos actuales de NGS:** Se identificó que el proceso de desarrollo de software en la organización se encuentra estructurado de manera funcional, pero con un alto grado de variabilidad entre proyectos y equipos. En términos generales, la construcción de software se realiza mediante compilaciones y empaquetados gestionados directamente por los desarrolladores, apoyados en repositorios de código como Git y en tableros de seguimiento de tareas. Sin embargo, estas actividades se llevan a cabo sin un pipeline

institucional de integración continua que defina etapas, responsables y criterios de avance.

Las pruebas se ejecutan principalmente de forma manual, tanto por desarrolladores como por personal de QA, dependiendo de la naturaleza del proyecto y de los tiempos disponibles. No se cuenta con un esquema formal de pruebas automatizadas que acompañe sistemáticamente cada cambio de código, lo que hace que la calidad dependa en gran medida de la experiencia de las personas y del tiempo asignado a cada entrega.

En cuanto al despliegue, la liberación de versiones hacia ambientes de prueba o producción se realiza de manera manual o semiautomática, con una fuerte dependencia de ciertos perfiles clave dentro de los equipos técnicos o de infraestructura. Los flujos de aprobación y paso entre ambientes no se encuentran documentados ni estandarizados a nivel organizacional, por lo que cada proyecto puede operar bajo acuerdos específicos con el cliente o el líder técnico.

En este contexto, intervienen principalmente los equipos de desarrollo, QA, soporte e infraestructura, pero su coordinación se apoya más en la comunicación informal y en la experiencia acumulada que en procesos formales y repetibles.

A continuación mostramos algunos de los problemas observados en el proceso actual:

**Figura 47** Problemas identificados en el proceso de desarrollo en NGS

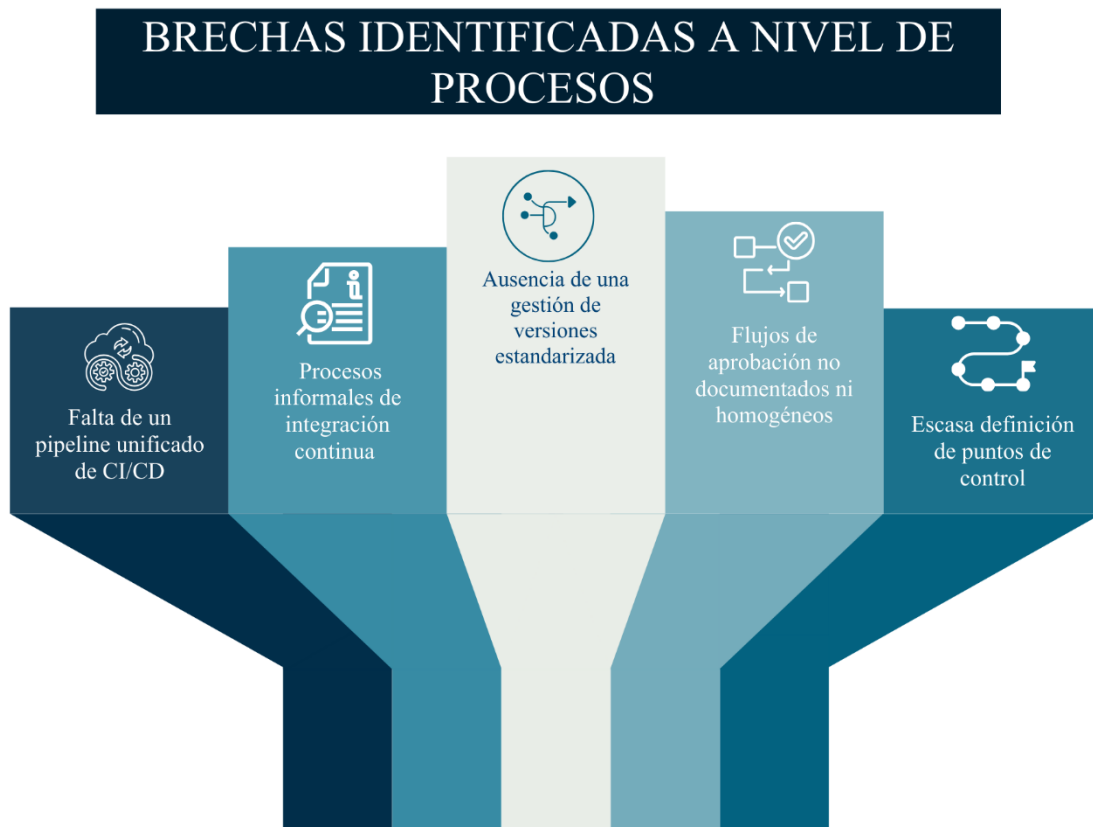


Nota: Elaboración propia.

Los instrumentos aplicados permitieron evidenciar que el proceso de desarrollo en NGS presenta una fuerte dependencia de tareas manuales, tanto en la compilación como en el empaquetado y los despliegues, lo que incrementa el riesgo de errores y limita la estandarización. Se identificó también la ausencia de lineamientos claros para la gestión de versiones, así como un predominio de pruebas manuales con poca automatización, afectando la rapidez y confiabilidad de las validaciones. La trazabilidad de los cambios es limitada y los ambientes presentan inconsistencias entre desarrollo, QA y producción, lo que genera fallas repetitivas y retrabajos derivados de errores humanos. En conjunto, estos problemas reflejan un proceso con carencias que dificultan avanzar hacia prácticas más automatizadas o modelos de integración y entrega continua.

A partir de estos problemas, se identificaron las siguientes brechas a nivel de proceso:

**Figura 48** Brechas principales en los procesos identificados en NGS



Nota: Elaboración propia.

Los resultados evidencian que el proceso actual de desarrollo en NGS no permite avanzar

hacia un modelo de integración continua debido a brechas fundamentales en su estructura operativa. La ausencia de un pipeline unificado, junto con procesos informales de integración, genera variabilidad entre equipos y limita la capacidad de repetir el proceso de forma consistente. A ello se suma la falta de una gestión estandarizada de versiones, lo que reduce la trazabilidad y dificulta controlar adecuadamente la evolución del código. Del mismo modo, los flujos de aprobación no documentados crean dependencias personales y cuellos de botella que afectan la velocidad de entrega.

Aparte, la escasa definición de puntos de control impide identificar fallos tempranamente, incrementando el riesgo de errores en etapas críticas como compilación, pruebas y despliegues. Estas brechas, en conjunto, condicionan la posibilidad de automatizar el proceso y establecer un pipeline confiable. El resultado es un flujo operativo que depende en gran medida de la intervención manual, carece de visibilidad integral y no garantiza la estabilidad necesaria para implementar prácticas robustas de CI/CD.

Las brechas identificadas indican que el proceso actual se encuentra en una fase previa a DevOps; las actividades existen, pero no están integradas dentro de un flujo continuo y automatizado. Esto significa que, aunque la organización cuenta con experiencia en desarrollo y despliegue, aún no dispone de la estructura de procesos necesaria para soportar un modelo de CI/CD estable y alineado con las buenas prácticas de la industria.

A continuación se presentan los principales hallazgos a nivel de procesos:

**Tabla 24** *Hallazgos principales en los procesos*

No	Hallazgo principal	Descripción breve
1	Procesos no estandarizados	El desarrollo, las pruebas y los despliegues varían entre equipos por falta de documentación formal.
2	Ausencia de pipeline CI/CD	Las actividades clave (compilación, pruebas, despliegue) continúan siendo manuales y no integradas.
3	Flujos y versiones informales	Las aprobaciones y la gestión de versiones dependen del criterio del equipo, afectando la trazabilidad.

4	Coordinación ad hoc entre áreas	La interacción entre desarrollo, QA e infraestructura no sigue procesos definidos y genera errores y retrabajos.
5	Madurez inicial de los procesos	Los procesos cumplen entregas mínimas, pero sin estructura suficiente para habilitar CI/CD.

Nota: Elaboración propia.

Estos descubrimientos ofrecen una base firme para explicar, en los próximos capítulos, la necesidad de crear un modelo de integración DevOps-Ágil-ITIL que mejore la estandarización, la automatización y la mejora continua en NGS.

**b) Brechas en herramientas y tecnología utilizadas en NGS:** A partir del inventario levantado por medio del Checklist y la encuesta aplicada al personal técnico, se identificó que NGS emplea un conjunto diverso de herramientas tecnológicas para el desarrollo y operación del software.

En la siguiente tabla se describen las herramientas que se utilizan actualmente en NGS:

**Tabla 25** *Inventario visual de herramientas actuales en NGS*

Categoría	Herramientas detectadas	Nivel de adopción	Observación
Control de versiones	Git, Azure DevOps Repos	Alta	Se utiliza de forma generalizada.
CI/CD	Azure DevOps Pipelines	Media-baja	No existe estandarización institucional.
Pruebas	Postman, SOAP UI	Media	Principalmente manuales.
Monitoreo	QlikView, Power BI	Baja	Uso aislado por equipos.
Contenedores	Docker, Kubernetes	Baja	Solo se han realizado pruebas piloto.

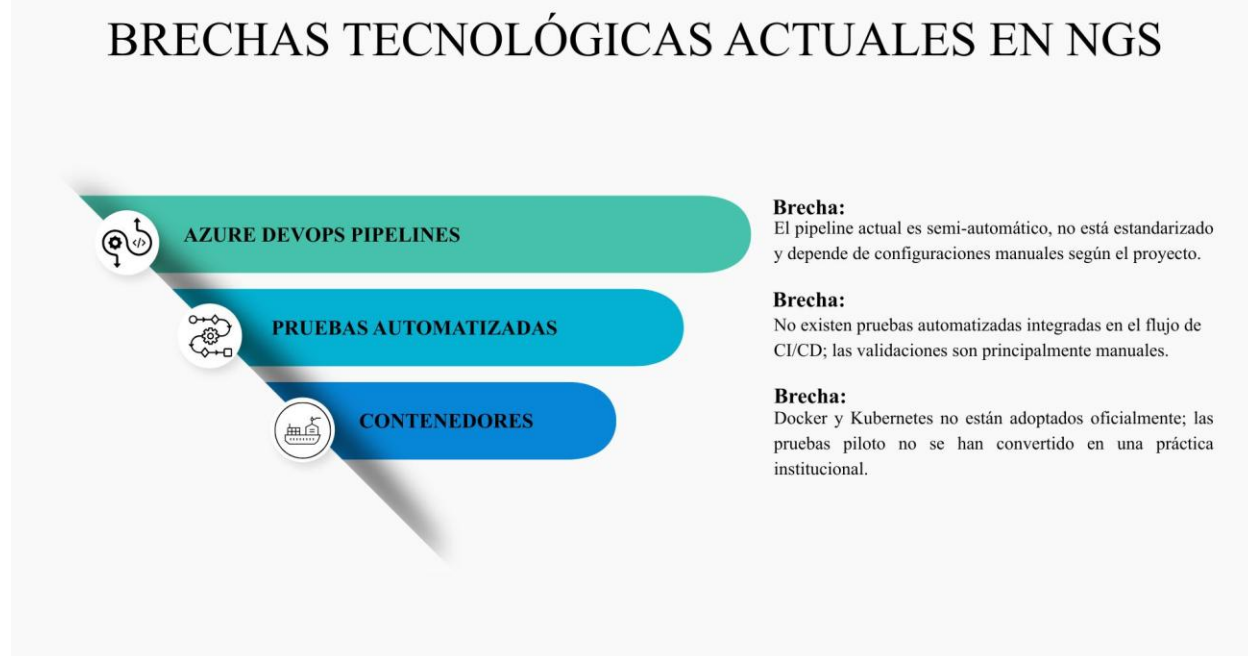
Nota: Elaboración propia basada en el Checklist técnico y en las secciones A y B de la encuesta.

Este inventario de herramientas actualmente utilizadas en NGS revela un ecosistema tecnológico funcional, pero fragmentado y con niveles de adopción desiguales entre equipos y proyectos. Aunque existen bases importantes como el uso extendido de Git y Azure DevOps Repos, otras áreas clave para habilitar un flujo CI/CD completo, como la automatización de

pruebas, el monitoreo integrado o la gestión de artefactos, presentan un uso parcial o incipiente. Esta heterogeneidad evidencia la necesidad de avanzar hacia una estandarización institucional que permita aprovechar plenamente las capacidades de las herramientas disponibles y sentar las bases para una integración continua más madura y sostenible.

Del análisis realizado se evidencian varias brechas importantes en el uso de herramientas dentro de NGS, principalmente relacionadas con la falta de estandarización, la adopción parcial de tecnologías clave y la limitada integración entre plataformas. Estas brechas afectan directamente la posibilidad de implementar un flujo CI/CD consistente. A continuación, se presentan las principales brechas detectadas:

**Figura 49** Brechas tecnológicas identificadas en NGS



Nota: Elaboración propia.

La revisión de los instrumentos permitió conectar las brechas detectadas con sus implicaciones directas sobre la capacidad de NGS para habilitar un flujo CI/CD maduro. En primer lugar, el uso semiautomatizado y no estandarizado de Azure DevOps Pipelines evidencia la falta de un servicio CI/CD homogéneo. Mientras las automatizaciones dependen del criterio y experiencia individual de los equipos, los procesos no pueden ser completamente automatizados ni gestionados de forma centralizada, lo que limita la trazabilidad y dificulta la implementación de

un flujo continuo confiable.

Asimismo, la ausencia de pruebas automatizadas impide establecer un proceso de integración y entrega continua, ya que no existe un mecanismo automático completamente implementado que valide la calidad del código en cada cambio realizado. Esto genera dependencia en validaciones manuales, aumenta el riesgo de errores y limita la velocidad de entrega.

Finalmente, la no adopción formal de contenedores como Docker o Kubernetes produce ambientes inconsistentes entre desarrollo, QA y producción. Al no contar con imágenes estandarizadas ni despliegues reproducibles, se incrementan los fallos por diferencias de configuración y se dificulta la escalabilidad de los entornos.

Derivado de este análisis, a continuación se presentan los principales hallazgos sobre las herramientas en NGS:

**Figura 50** Principales hallazgos sobre las herramientas en NGS



Nota: Elaboración propia.

- c) **Brechas en competencias y capacidades:** Los instrumentos aplicados evidencian que el equipo técnico de NGS posee una base sólida de conocimiento en desarrollo de software, uso de repositorios de código y manejo de herramientas como Git y Azure DevOps. El personal demuestra experiencia práctica en la construcción de aplicaciones, resolución de

incidencias y trabajo colaborativo dentro de metodologías ágiles, aunque estas últimas aún se encuentran en desarrollo y sin roles formales institucionalizados.

Sin embargo, cuando se evalúan las competencias específicas necesarias para un modelo DevOps y para habilitar un flujo CI/CD, se observa una brecha significativa. La mayoría del personal no ha tenido experiencia directa con automatización de pruebas, pipelines, herramientas de integración continua, contenedores o infraestructura como código. El conocimiento existe a nivel conceptual, pero no se refleja en prácticas operativas dentro de los proyectos actuales.

A continuación se detallan las principales brechas a nivel de competencias en los equipos de desarrollo:

**Figura 51** Brechas principales de competencias identificadas



Nota: Elaboración propia.

Esto refleja que aunque NGS cuenta con talento técnico con experiencia en desarrollo y QA, las competencias específicas requeridas para implementar DevOps y CI/CD aún no están consolidadas.

**Tabla 26** *Competencias actuales y brechas identificadas*

<b>Categoría</b>	<b>Situación Actual Identificada</b>	<b>Brechas de Competencia Detectadas</b>
<b>Prácticas DevOps generales</b>	La encuesta indica nivel variable de familiaridad con DevOps (84.6% dice conocer el término, pero la aplicación es desigual).	Conocimiento conceptual, pero limitado en la aplicación práctica dentro de los proyectos.
<b>Automatización de procesos</b>	El Checklist y la encuesta muestran predominio de actividades manuales (compilación, pruebas, despliegue).	Falta de competencias para automatizar integraciones, pruebas y despliegues.
<b>Pruebas automáticas</b>	Existe casi nula implementación de pruebas automáticas; prevalece el enfoque manual.	Ausencia de habilidades para diseñar, configurar y ejecutar pruebas automatizadas.
<b>Uso de pipelines en Azure DevOps</b>	El pipeline existe, pero no está estandarizado ni se aplica de forma uniforme.	Competencias limitadas para estructurar y mantener pipelines consistentes.
<b>Contenedores y virtualización</b>	Experiencia inicial o piloto con Docker; Kubernetes no se usa.	Falta de conocimiento práctico en creación y uso de contenedores.

Nota: Elaboración propia.

En conjunto, los instrumentos aplicados muestran que el equipo cuenta con conocimientos generales sobre prácticas DevOps y tecnologías asociadas, pero su aplicación es aún limitada. Estas competencias deben fortalecerse para que NGS pueda habilitar de forma efectiva un futuro modelo de CI/CD.

Según el análisis realizado a nivel de competencias en NGS, hemos identificado los siguientes hallazgos:

**Figura 52** Hallazgos principales sobre competencias



Nota: Elaboración propia.

Los resultados obtenidos a partir de la encuesta, el Checklist técnico y el análisis FODA permitieron consolidar una visión integral sobre las brechas actuales de NGS en procesos, herramientas y competencias. Al analizar conjuntamente estos elementos, se identificaron patrones que se refuerzan entre sí y que explican gran parte de las limitaciones para avanzar hacia un modelo de integración y entrega continua. La siguiente tabla resume los hallazgos principales, evidenciando cómo cada dimensión impacta y condiciona a las demás:

**Tabla 27** Resumen consolidado de hallazgos en procesos, herramientas y competencias

Dimensión	Hallazgo principal	Descripción	Relación con otras dimensiones
Procesos	Procesos no estandarizados	El desarrollo, las pruebas y los despliegues varían entre equipos y no siguen lineamientos formales.	Se refleja en el uso inconsistente de herramientas y en la falta de competencias uniformes.
	Ausencia de pipeline CI/CD	Actividades clave (build, pruebas, despliegue) siguen siendo manuales o parciales.	Depende de herramientas no adoptadas y de habilidades insuficientes en automatización.
	Flujos y versiones informales	La aprobación y el versionamiento dependen del criterio de cada equipo, afectando la trazabilidad.	La falta de dominio en GitFlow y prácticas formales limita su estandarización.

	Coordinación ad hoc entre áreas	La interacción entre desarrollo, QA e infraestructura no sigue procesos definidos.	La carencia de herramientas integradas y de roles especializados amplifica esta debilidad.
<b>Herramientas</b>	Inexistencia de pipeline estandarizado	No se cuenta con un flujo automatizado para construcción, pruebas y despliegue.	Impide madurar los procesos y exige mayor nivel técnico del equipo.
	Ausencia de herramientas de pruebas automatizadas	Las pruebas son manuales, sin frameworks institucionalizados.	Requiere formación especializada en QA Automation y scripting.
	Baja adopción de contenedores	Docker y Kubernetes solo se han probado de forma limitada.	Limita la posibilidad de ambientes reproducibles y exige nuevas competencias.
<b>Competencias</b>	Falta de habilidades para automatización	El equipo requiere capacitación en CI/CD, scripting y automatización de tareas.	Dificulta la estandarización del pipeline y la adopción de herramientas.
	Ausencia de experiencia en pruebas automatizadas	El equipo no domina frameworks de testing automatizado.	Se traduce en dependencia de pruebas manuales y retrasos en el ciclo.
	Bajo dominio de contenedores y orquestación	No hay experiencia sólida en Docker ni Kubernetes.	Impide trabajar con ambientes consistentes y limita la evolución arquitectónica.
	Necesidad de roles especializados	No existen perfiles como DevOps Engineer o QA Automation.	Impacta directamente la madurez del proceso y la operación del pipeline.

Nota: Elaboración propia.

En conjunto, estos hallazgos evidencian que las brechas de NGS no se comportan de manera aislada, sino como un sistema interdependiente donde las debilidades en procesos, herramientas y competencias se retroalimentan mutuamente. Esta visión consolidada permite comprender con claridad la situación actual y constituye la base para orientar las acciones del modelo propuesto, priorizar intervenciones y avanzar hacia un entorno de desarrollo más estandarizado, automatizado y alineado con prácticas modernas de DevOps.

#### 4.2.2 IDENTIFICACIÓN DE PRÁCTICAS Y HERRAMIENTAS DEVOPS APLICABLES EN NGS Y SU ALINEACIÓN CON ITIL V4

Para avanzar en la identificación de las prácticas y herramientas DevOps que podrían incorporarse de manera efectiva en NGS, así como su alineación con ITIL V4, se aplicaron distintos instrumentos que permiten comprender cómo opera actualmente la organización y qué tan preparada se encuentra para adoptar estas prácticas. Estos insumos ofrecen una visión amplia

sobre la colaboración entre equipos, el nivel de automatización existente, la integración de procesos y el grado de madurez de las prácticas ITIL que sirven de soporte a DevOps. La información obtenida constituye la base para analizar, de forma integral y contextualizada, qué elementos son factibles de implementar en NGS según su realidad actual y las buenas prácticas reconocidas en la industria.

A continuación, se muestran los instrumentos aplicados para el objetivo 2:

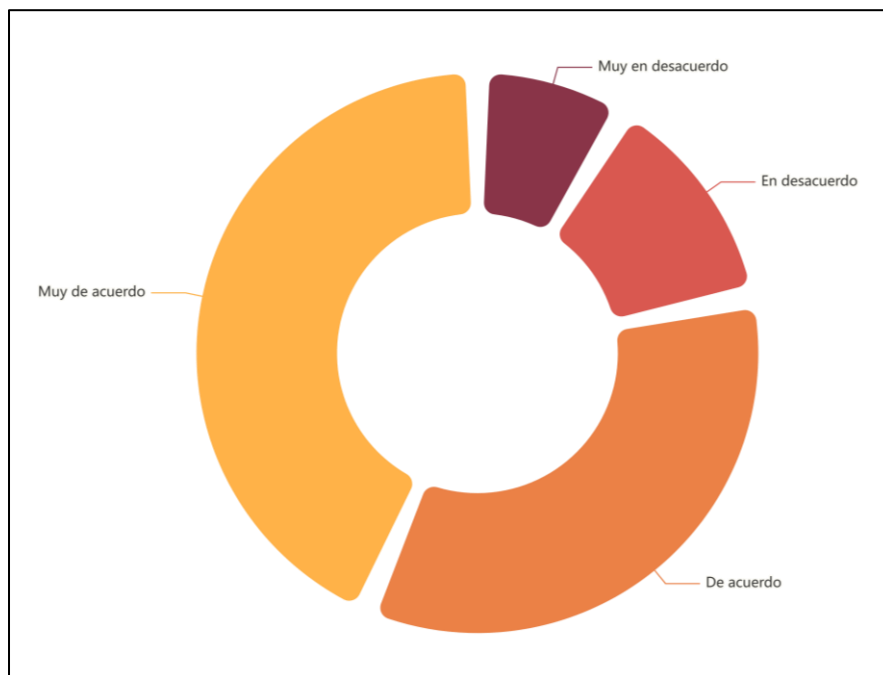
#### **ENCUESTA - ALINEACIÓN CON ITIL V4**

En la encuesta de Alineación con ITIL V4 se observa que, si bien existe conocimiento básico sobre ITIL V4, su implementación en NGS aún no es formal ni sistemática. Prácticas como gestión de cambios, incidentes y despliegues se aplican parcialmente, sin métricas, roles definidos o procesos documentados. Los resultados indican que NGS podría beneficiarse al incorporar ITIL V4 como marco de estandarización y trazabilidad, especialmente para fortalecer la alineación entre desarrollo, QA y operaciones.

A continuación, se presentan y describen las preguntas y respuestas desarrolladas:

1. Los cambios en los sistemas se registran y aprueban formalmente antes de su implementación.

**Figura 53** *Gestión formal de cambios*



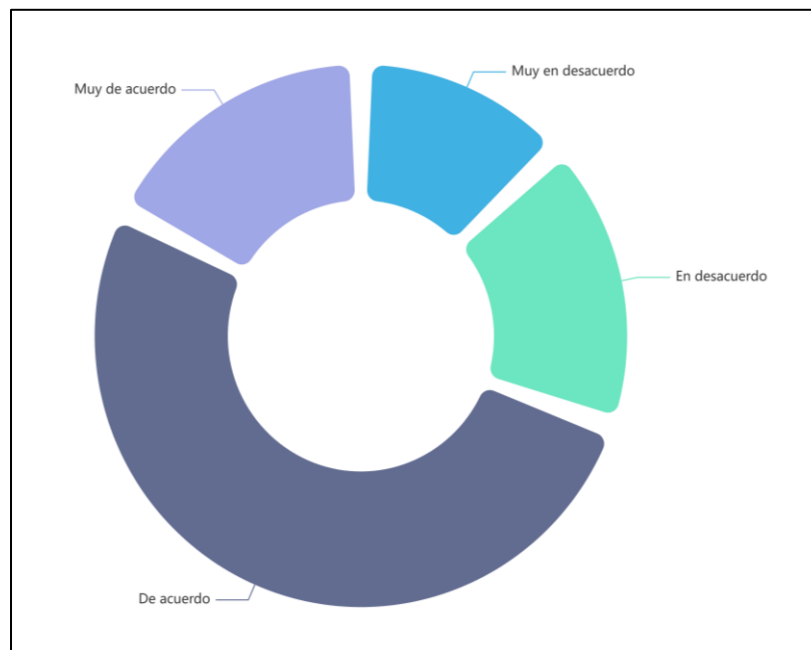
Nota: Elaboración propia.

La mayoría de los colaboradores manifestó estar Muy de acuerdo con que los cambios se registran y aprueban formalmente, es decir, que los cambios se registran, revisan y aprueban antes de su implementación. Esta percepción refleja que existe una práctica establecida y reconocida por gran parte del personal, lo cual contribuye a mantener el control y la trazabilidad en el proceso CI/CD.

El predominio de percepciones positivas respecto al registro y aprobación formal de cambios indica que NGS ya utiliza prácticas compatibles con *Change Enablement* de ITIL V4. Esta estructura constituye una base favorable para integrar mecanismos DevOps como control de versiones, flujos automatizados de aprobación, validaciones previas y auditorías integradas en el pipeline CI/CD. En este sentido, el proceso actual permite que las prácticas DevOps se incorporen sin fricciones, aprovechando un marco ya conocido por los equipos.

2. Los incidentes se gestionan con tiempos de respuesta definidos y acordados (SLA).

**Figura 54** *Gestión de incidentes con SLA*



Nota: Elaboración propia.

Una parte de los colaboradores tiene percepción dividida sobre la gestión de incidentes con tiempos de respuesta establecidos (SLA). Una parte del equipo indica acuerdo o muy de acuerdo

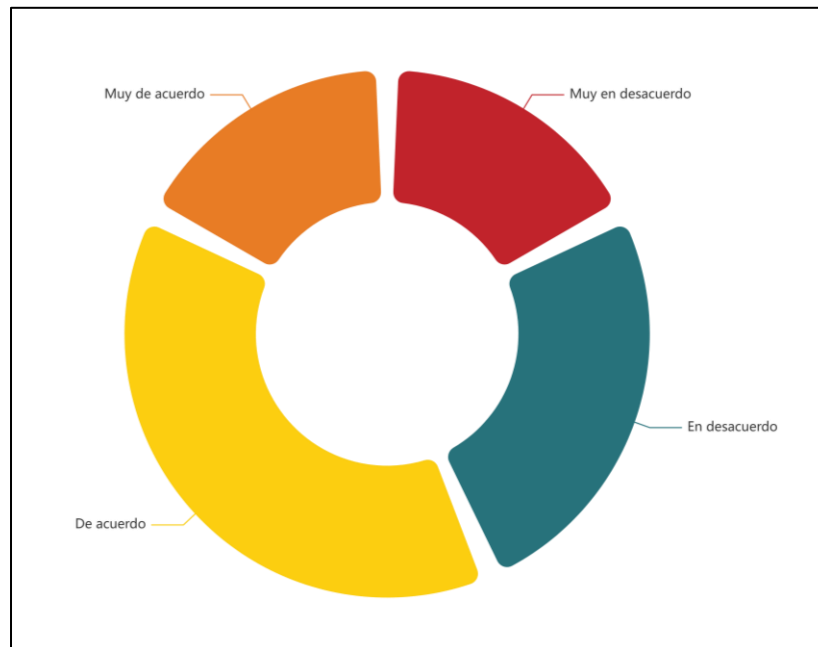
respecto a que los incidentes se atienden siguiendo tiempos definidos, lo que sugiere que sí existe una estructura de atención y priorización.

Otra parte de los colaboradores igualmente significativa expresa desacuerdo o muy en desacuerdo, lo que indica que, para muchos, los tiempos de respuesta no se cumplen de manera consistente o no están claramente definidos para todos los casos.

La percepción general de que existen tiempos de respuesta establecidos sugiere que NGS ya maneja criterios operativos alineados con la gestión de incidentes propuesta por ITIL V4. Esto crea una oportunidad para integrar prácticas DevOps vinculadas con observabilidad, monitoreo continuo, métricas operativas y automatización de alertas. La presencia de SLA facilita la incorporación de herramientas DevOps destinadas a mejorar la confiabilidad del servicio y a medir indicadores clave como tiempo de recuperación o frecuencia de incidentes.

### 3. Existe trazabilidad entre los cambios implementados y los incidentes generados.

**Figura 55** *Trazabilidad entre cambios e incidentes*



Nota: Elaboración propia.

Las opiniones sobre la trazabilidad entre cambios e incidentes también muestran división. Una proporción considerable del personal indica acuerdo o muy de acuerdo, lo que refleja que, para ellos, existe una relación documentada y visible entre ambos procesos. Una parte relevante

expresa desacuerdo o muy en desacuerdo, señalando que dicha trazabilidad no siempre está presente, es incompleta o depende del equipo responsable.

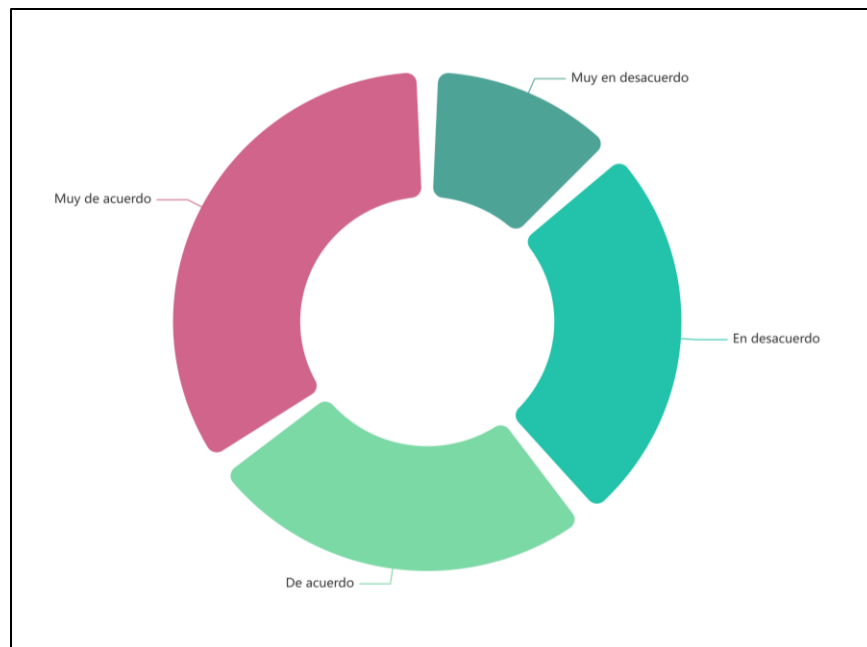
A pesar de las diferentes opiniones, más de la mitad de los participantes menciona que hay trazabilidad. Esto sugiere que NGS ya tiene cierta conexión entre cambios e incidentes, lo cual es necesario para combinar herramientas de pipeline CI/CD con la gestión ITIL. Esta trazabilidad inicial puede potenciarse incorporando prácticas DevOps como:

- Integración código-incidentes mediante Git/Azure DevOps
- Automatización de auditorías de cambios
- Seguimiento de incidentes derivados de deployments
- Uso de tableros con métricas DORA

De esta forma, el proceso puede convertirse en un sistema integrado Agile-DevOps-ITIL, donde cada cambio tenga impacto medible sobre la estabilidad del servicio.

4. Los despliegues siguen flujos o plantillas aprobadas por el área de gestión de servicios.

**Figura 56** *Uso de flujos o plantillas aprobadas para despliegues*



Nota: Elaboración propia.

El equipo de colaboradores muestra percepciones mixtas respecto al uso de flujos o

plantillas aprobadas para despliegues. Una parte significativa indica acuerdo o muy de acuerdo, evidenciando que sí existen lineamientos y plantillas que guían el proceso de despliegue en varios casos.

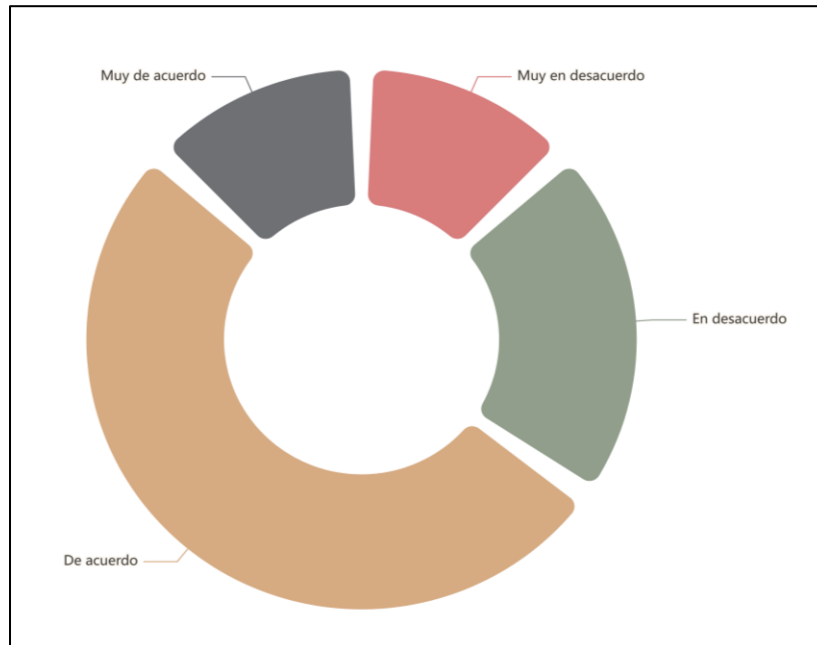
Los resultados evidencian que existe un esfuerzo organizacional por estructurar los despliegues, lo cual coincide con las mejores prácticas de ITIL V4 (Release Management). Esta estructura constituye una oportunidad directa para:

- Integrar pipelines CI/CD basados en plantillas
- Definir flujos automatizados de aprobación
- Estandarizar los despliegues en todos los equipos
- Integrar DevOps con la gestión de servicios ITIL
- Automatizar validaciones previas mediante pruebas continuas y escaneos de calidad

La aceptación de flujos ya existentes facilita la incorporación de prácticas DevOps sin necesidad de cambios radicales en la cultura actual.

5. Se mantienen registros históricos de incidentes, cambios y versiones de software.

**Figura 57** Registros históricos de incidentes, cambios y versiones



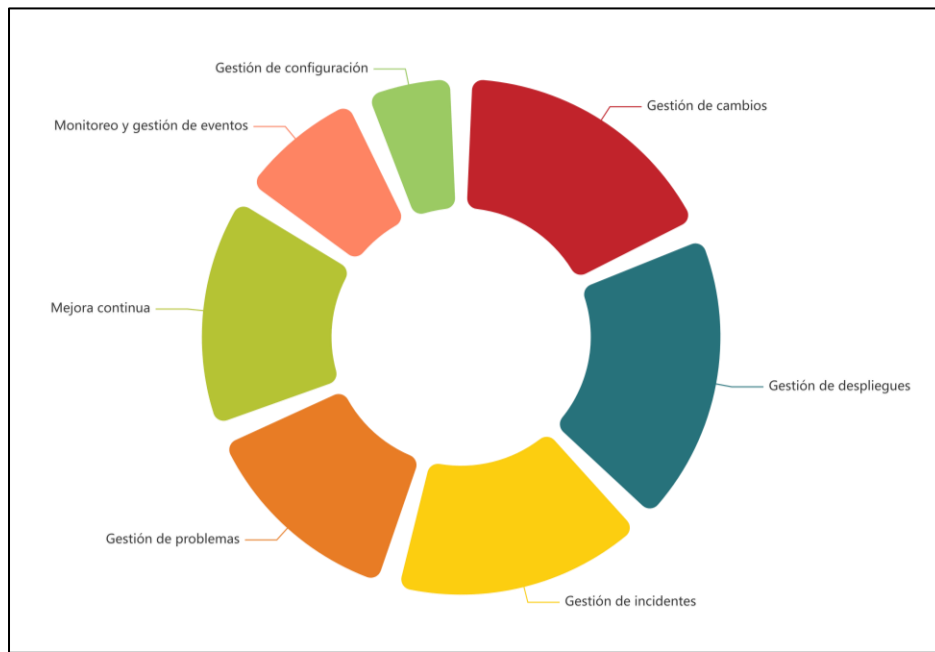
Nota: Elaboración propia.

La mayoría de los colaboradores indica acuerdo o muy de acuerdo con que se mantienen registros históricos de incidentes, cambios y versiones. Esto sugiere que la organización cuenta con prácticas de documentación que permiten conservar evidencia histórica y dar seguimiento a eventos relevantes del proceso.

La existencia de registros históricos es fundamental para incorporar herramientas DevOps enfocadas en auditoría, versionamiento automatizado, trazabilidad avanzada y análisis de tendencias. Estos registros habilitan la integración de métricas DevOps y de buenas prácticas ITIL V4, permitiendo medir estabilidad, calidad y desempeño del ciclo de vida del software de manera más efectiva.

6. ¿Qué prácticas ITIL considera mejor implementadas en NGS?

**Figura 58** Prácticas ITIL consideradas mejor implementadas en NGS



Nota: Elaboración propia.

Los resultados evidencian que NGS cuenta con varias prácticas ITIL v4 relativamente consolidadas, especialmente en áreas como la gestión de despliegues, la gestión de cambios y el manejo de incidentes. Estas prácticas son reconocidas por buena parte del equipo como procesos formales que contribuyen a mantener control, estabilidad y coordinación en los servicios, constituyendo una base sólida para introducir prácticas DevOps sin generar fricción.

La gestión de problemas también es muy importante, lo que sugiere que se están haciendo esfuerzos para analizar las causas principales y hacer mejoras de manera continua, siguiendo el enfoque de aprendizaje iterativo propio de DevOps. La gestión de configuración y activos es la práctica con menor valoración. Esto sugiere un área importante de fortalecimiento, dado que esta función es clave para automatizar procesos avanzados, asegurar la trazabilidad de versiones y mantener control del entorno tecnológico. Aunque su presencia es limitada, representa un punto de partida sobre el cual podría construirse una base más robusta para soportar iniciativas de automatización y despliegue continuo. Sin embargo, su presencia, aunque limitada, permite contar con una base inicial sobre la cual construir.

7. Desde su experiencia, ¿Qué aspecto o práctica relacionada con la gestión de servicios de TI (por ejemplo, control de cambios, incidentes o despliegues) considera más importante fortalecer para que se alinee mejor con las prácticas DevOps? Explique brevemente su respuesta.

**Tabla 28** Descripción y análisis de las respuestas abiertas

Aspecto clave	Descripción breve	Análisis
<b>Gestión de incidentes</b>	Se solicita mejorar la documentación, seguimiento y resolución de incidentes para evitar recurrencia de fallas.	Una gestión de incidentes más estructurada mejora la estabilidad operativa y facilita la integración con prácticas DevOps e ITIL V4.
<b>Gestión de cambios</b>	Es el aspecto más mencionado; se percibe lento, manual y con poca trazabilidad.	Es crítico para alinearse con DevOps, ya que requiere automatización, documentación y gobernanza para permitir entregas continuas.
<b>Monitoreo y gestión de eventos</b>	Se señala que el monitoreo es limitado y reactivo.	Un monitoreo proactivo es esencial para DevOps e ITIL V4, permitiendo detectar problemas temprano y mejorar la calidad del servicio.
<b>Gestión de despliegues</b>	Se observan oportunidades para mejorar plantillas, flujos y automatización.	Optimizar esta práctica facilita despliegues más seguros, frecuentes y alineados con pipelines CI/CD.

<b>Gestión de problemas</b>	Se menciona para evitar repetición de errores recurrentes.	Abordar causas raíz reduce incidentes y mejora la estabilidad del software, favoreciendo la mejora continua.
<b>Documentación y trazabilidad</b>	Se destaca la falta de documentación en varios procesos.	Es fundamental para controlar cambios, medir impacto y garantizar alineación entre DevOps e ITIL V4.

Nota: Elaboración propia basada en la pregunta 7 de la sección C de la encuesta.

## MATRIZ DE MADUREZ DE PRÁCTICAS ITIL V4 EN NGS

**Tabla 29** *Matriz Consolidada de Madurez de Prácticas ITIL V4 en NGS*

No	Práctica ITIL v4	Nivel esperado	Nivel actual predominante en NGS	Brecha cualitativa	Justificación
1	Gestión de Cambios	Nivel 3 - Definido / Nivel 4 - Cuantitativo	Nivel 4- Cuantitativo con casos aislados en Nivel 2 - Gestionado	Brecha leve	La práctica está relativamente cerca del nivel esperado, pero la aplicación aún no es uniforme. Aunque algunos proyectos poseen cambios controlados, documentados y validados, otros siguen siendo reactivos y con poca trazabilidad. Falta estandarización general.
2	Gestión de Despliegues	Nivel 3 - Definido / Nivel 4 - Cuantitativo	Varía entre Nivel 2 - Gestionado, Nivel 3 - Definido y algunos casos Nivel 4 - Cuantitativo	Brecha moderada	Los despliegues muestran una madurez desigual. Existen pipelines y automatización en ciertos equipos, pero otros dependen de procesos manuales, poca documentación y diferencias entre entornos y clientes. Requiere consolidación y uniformidad.
3	Gestión de Incidentes	Nivel 2 - Gestionado / Nivel 3 - Definido	Predomina Nivel 3 - Definido	Brecha leve	En general se atienden incidentes con formalidad, pero persisten respuestas reactivas, tiempos no cumplidos y falta de documentación consistente. Se requiere fortalecer el registro y la estandarización del proceso.

4	Gestión de Problemas	Nivel 2 - Gestionado / Nivel 3 - Gestionado	Entre Nivel 2 - Gestionado y Nivel 3 - Definido	Brecha leve a moderada	Aunque existe análisis de problemas, no hay metodologías formales para estudiar la causa raíz ni repositorios claros. El proceso se ejecuta de forma parcial, dependiente de cada equipo.
5	Mejora Continua	Nivel 3 - Definido / Nivel 4 - Cuantitativo	Entre Nivel 2 - Gestionado y Nivel 3 - Definido	Brecha moderada a alta	Es una de las prácticas más débiles. Las mejoras se realizan de forma empírica, sin métricas, ciclos formales ni mecanismos institucionales. Falta una estructura metodológica para asegurar mejoras constantes.
6	Monitoreo y Gestión de Eventos	Nivel 3 - Definido	Entre Nivel 2 - Gestionado y Nivel 3 - Definido	Brecha leve a moderada	El monitoreo existe, pero no de forma integrada ni automatizada. Falta robustez, alertas oportunas y visibilidad completa de eventos. La práctica es irregular dependiendo del proyecto o cliente.
7	Gestión de Configuración y Activos	Nivel 3 - Definido	Level 2 - Gestionado	Brecha moderada a alta	Es una de las prácticas con menor madurez. No existe un inventario formal, control de configuraciones consistente o repositorios centralizados. Hay desconocimiento general sobre el proceso.

Nota: Elaboración propia.

**Tabla 30 Niveles ITIL**

Nivel	Nombre del Nivel	Descripción breve
Nivel 1	Inicial	El trabajo se realiza, pero no siempre como resultado del SVS; existe baja estandarización.
Nivel 2	Gestionado	Los objetivos del SVS se logran repetidamente, pero los procesos aún no están estandarizados.
Nivel 3	Definido	Existen estándares organizados y documentados que sirven como guía formal.
Nivel 4	Cuantitativo	Los resultados del SVS se miden y evalúan para la toma de decisiones.
Nivel 5	Optimizando	La práctica se mejora continuamente y se optimiza de manera sistemática.

Nota: Elaboración propia basada en el modelo de madurez de PeopleCert.

Con el propósito de encontrar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma satisfactoria al marco ágil existente en NGS y considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI, se llevó a cabo una revisión documental. Lo cual permitió recopilar y analizar información especializada en las diferentes fuentes académicas, marcos oficiales de ITIL 4 y DevOps, así como guías de proveedores relevantes como AWS, Atlassian, Microsoft, entre otros. La revisión fue desarrollada tomando en cuenta los resultados de los instrumentos aplicados previamente, permitiendo contrastar el estado actual de NGS con las prácticas reconocidas internacionalmente, y establecer puntos de compatibilidad, viabilidad de adopción y rutas de alineación entre enfoques DevOps, ITIL y ágil (Scrum).

Para este análisis también se seccionó en 3 grandes dimensiones, como prácticas, herramientas y políticas, las cuales se describen a continuación:

- **Prácticas DevOps aplicables en NGS:** Aquí se muestran las prácticas que consideramos pertinentes aplicar en NGS y, de igual forma, realizamos un análisis de compatibilidad entre Agile-DevOps-ITIL 4.

**Figura 59** Prácticas DevOps identificadas para aplicar en NGS



Nota: Elaboración propia.

La implementación de las prácticas DevOps identificadas para NGS responde directamente a las brechas encontradas en el diagnóstico del objetivo 1. La ausencia de un pipeline CI/CD institucionalizado, la variabilidad en los procesos entre equipos y el predominio de pruebas manuales muestran una necesidad urgente de adoptar prácticas como integración y entrega continua, automatización de pruebas y control de versiones formal. Estas permitirían estandarizar la forma en que se compila, valida y despliega el software, reduciendo retrabajos, errores humanos y cuellos de botella.

La incorporación de Infraestructura como Código (IaC), monitoreo y retroalimentación técnica, y gestión colaborativa del flujo permitiría mitigar la falta de ambientes consistentes, mejorar la visibilidad operativa y fomentar la toma de decisiones basada en datos. Esto responde al hallazgo de una madurez inicial en los procesos, donde actualmente no existen mecanismos formales para validar continuamente la calidad del producto ni para controlar la configuración de la infraestructura. Sin embargo, para que estas prácticas se implementen efectivamente, NGS debe priorizar el cierre de brechas de competencias en automatización, scripting, uso de contenedores y cultura de mejora continua. Su adopción, más allá de herramientas, requiere habilitar capacidades internas alineadas con un enfoque colaborativo, técnico y orientado a flujos de valor.

Es importante mencionar que DevOps y Scrum son marcos compatibles que, al integrarse, permiten combinar la agilidad en la gestión de proyectos con la eficiencia técnica en la entrega continua. Mientras Scrum estructura el trabajo colaborativo y enfocado en valor de negocio, DevOps aporta automatización, monitoreo y despliegue continuo, fortaleciendo el ciclo completo de desarrollo con prácticas técnicas modernas. A continuación se muestra una tabla en la cual se muestran los elementos que presentan una relación entre Scrum y el enfoque DevOps:

**Tabla 31** *Compatibilidad y complementariedad entre Scrum y DevOps*

Elemento	Enfoque en Scrum	Enfoque en DevOps	Compatibilidad Potencial
<b>Planificación</b>	Sprint Planning, backlog priorizado	Planificación de pipelines y flujos de integración	DevOps puede integrarse como parte del DoD técnico
<b>Construcción (Build)</b>	Implementación dentro del Sprint	Automatización del build con scripts y CI	CI puede ejecutarse dentro del Sprint
<b>Pruebas (Test)</b>	Validación funcional manual	Pruebas automatizadas y regresión continua	Scrum define qué probar, DevOps cómo hacerlo
<b>Entrega (Release)</b>	Al final del Sprint	Entrega continua automatizada (CD)	CD puede integrarse al cierre de cada Sprint

<b>Retroalimentación</b>	Retrospectiva + Feedback de stakeholders	Métricas, monitoreo y alertas	Se pueden integrar indicadores técnicos en las reuniones
<b>Roles técnicos</b>	PO, SM, Development Team	QA Automation, DevOps Engineer	DevOps complementa el equipo Scrum con capacidades clave
<b>Flujo de trabajo</b>	Iterativo e incremental	Automatizado, continuo	Flujo ágil puede incorporar pipelines y prácticas CI/CD

Nota: Elaboración propia.

También se identificaron 7 prácticas de las 34 de ITIL V4 para alinearlas con DevOps, la madurez de estas prácticas en NGS fue evaluada a través del instrumento matriz de madurez. A continuación se enumeran dichas prácticas:

- Gestión de cambios
- Gestión de despliegues
- Gestión de incidentes
- Gestión de problemas
- Mejora continua
- Monitoreo y Gestión de Eventos
- Gestión de Configuración y Activos

La alineación entre las prácticas DevOps y las siete prácticas de ITIL V4 identificadas en NGS refleja una alta complementariedad técnica y operativa. DevOps aporta automatización, agilidad y enfoque en la entrega continua, mientras que ITIL V4 establece marcos estructurados para la gestión y control de servicios de TI. Por ejemplo, la gestión de cambios e incidentes en ITIL se fortalece con pipelines de CI/CD y despliegues controlados de DevOps; la mejora continua encuentra soporte en métricas y feedback automatizado; y prácticas como gestión de activos y monitoreo se integran naturalmente con IaC y observabilidad en tiempo real. La madurez evaluada en estas prácticas en NGS evidencia un punto de partida sólido, aunque aún limitado, lo cual abre la oportunidad para aplicar DevOps como catalizador de modernización en estos procesos tradicionales.

A continuación se muestra un mapeo entre DevOps, Scrum e ITIL V4, el cual demuestra que es posible alinear estos 3 enfoques en la organización:

**Tabla 32** Matriz de mapeo entre DevOps, Scrum e ITIL V4

Elemento	DevOps	Scrum	ITIL v4
<b>Cambio</b>	Automatizado, frecuente, impulsado por CI/CD	Iterativo, basado en priorización del backlog	Controlado, aprobado y documentado mediante gestión de cambios
<b>Entrega</b>	Entrega continua (CD) mediante pipelines automatizados	Entregas incrementales al final de cada Sprint	Planificación detallada y entregas basadas en diseño de servicios
<b>Retroalimentación</b>	Monitoreo continuo, alertas automáticas, métricas en tiempo real	Retrospectivas, feedback del cliente y del equipo	Mejora continua dentro del ciclo de vida del servicio
<b>Roles</b>	DevOps Engineer, QA Automation, SRE, Platform Engineer	Product Owner (PO), Scrum Master (SM), Development Team	Dueños de procesos, Gestores de cambios, Coordinadores de liberaciones
<b>Infraestructura</b>	Infraestructura como código (IaC), contenedores (Docker), orquestación (K8s)	Soporte limitado, no detalla infraestructura técnica	Gestión de configuración, gestión de activos de TI
<b>Automatización</b>	Fuerte enfoque: pipelines, pruebas, integración y despliegue	No incluida de forma nativa	Apoiada en la gestión técnica, pero no es su enfoque principal
<b>Visibilidad</b>	Dashboards, logs centralizados, trazabilidad del pipeline	Tableros de tareas (Kanban, burndown), reuniones diarias	Reportes de indicadores de servicio y cumplimiento de SLA
<b>Calidad</b>	Asegurada mediante pruebas automatizadas y monitoreo constante	Revisión en Sprint y pruebas manuales	Control de calidad vía gestión del cambio y validación del servicio
<b>Colaboración</b>	Fomenta colaboración entre Dev, QA y Ops (modelo cross-funcional)	Trabajo conjunto diario, énfasis en equipo autoorganizado	Integración entre funciones, pero roles más jerárquicos
<b>Gestión del conocimiento</b>	Documentación automatizada, wikis, versionamiento en repositorios	Compartido en retrospectivas y refinamientos	Gestión formal del conocimiento y de activos del servicio

Nota: Elaboración propia.

Luego de realizar este análisis de compatibilidad entre Scrum-DevOps-ITIL V4, se encontraron los siguientes hallazgos a nivel de prácticas:

**Tabla 33** Hallazgos principales sobre las prácticas identificadas para NGS

Categoría	Hallazgo	Instrumento
<b>Prácticas ITIL parcialmente maduras</b>	La madurez actual de ITIL en NGS muestra potencial en cambios e incidentes, pero debilidades serias en monitoreo y configuración.	Encuesta Sección C, Matriz ITIL

<b>Compatibilidad entre Scrum-DevOps-ITIL V4</b>	Existen puntos de sinergia entre DevOps, Scrum e ITIL v4 que pueden aprovecharse con procesos formales y automatización.	Revisión Documental
<b>Desalineación con marcos ágiles y DevOps</b>	Aunque las prácticas tienen potencial de alineación, su implementación actual no dialoga de forma efectiva con los sprints, retrospectivas o flujos automatizados.	

Nota: Elaboración propia.

Las prácticas DevOps identificadas a muestran una alta compatibilidad tanto con el marco ágil Scrum como con las prácticas ITIL v4 seleccionadas. Este mapeo evidencia que DevOps puede actuar como un puente integrador, complementando la flexibilidad e iteración de Scrum con la estructura y control de ITIL, permitiendo así estandarizar y automatizar procesos clave del desarrollo y la gestión de servicios. Esta sinergia resulta especialmente relevante para NGS, dada su necesidad de modernizar flujos sin perder gobernabilidad.

- **Herramientas identificadas para una mejor implementación de DevOps en NGS:** Esta actividad se puede relacionar con la práctica DevOps “empoderar a los equipos para elegir herramientas”. En este sentido, esta sección presenta las herramientas identificadas como más adecuadas para NGS, tomando en cuenta los hallazgos del diagnóstico previo, las necesidades del ciclo CI/CD y su alineación con las prácticas DevOps. Estas herramientas buscan mejorar la integración, las pruebas, el despliegue, el monitoreo y la trazabilidad de los procesos de desarrollo y operaciones. A continuación se muestran las herramientas que podrían colaborar a una mejor implementación de las prácticas DevOps en NGS:

**Tabla 34** *Herramientas DevOps identificadas*

<b>Herramienta</b>	<b>Función principal</b>	<b>Relevancia para NGS</b>
Git/GitLab/GitHub	Control de versiones	Ya usada en NGS
Jenkins / Azure Pipelines	Orquestación CI/CD	Uso parcial actual
SonarQube	Análisis de calidad del código	Recomendado
Docker / Compose	Contenedores y ambientes consistentes	Baja adopción actual
Kubernetes	Orquestación de contenedores	Requiere capacitación
Ansible / Terraform	Infraestructura como código	No utilizados aún
Grafana / Prometheus	Monitoreo e indicadores	Requiere integración

Nota: Elaboración propia.

La implementación de un conjunto adecuado de herramientas DevOps es esencial para lograr una integración continua (CI) y entrega continua (CD) eficiente, alineada con el ciclo completo DevOps (Plan-Code-Build-Test-Release-Deploy-Operate-Monitor). Herramientas como Git y Azure DevOps permiten gestionar versiones y tareas de desarrollo con trazabilidad, mientras que Jenkins y pipelines automatizados permiten compilar, probar y desplegar sin intervención manual. Docker ofrece consistencia entre entornos mediante contenedores, y herramientas como SonarQube aseguran la calidad del código de forma automatizada. Cuando estas herramientas se integran en un flujo orquestado, se facilita la detección temprana de errores, se reduce el tiempo de entrega y se incrementa la resiliencia de los despliegues.

En términos de viabilidad, NGS ya ha dado pasos iniciales en el uso de herramientas clave como Git y Azure DevOps, lo que facilita su expansión. Jenkins, aunque no estandarizado, es familiar para algunos equipos. Herramientas como Docker y SonarQube presentan viabilidad media, pues requieren formación técnica, pero no implican grandes inversiones. Por otro lado, soluciones más avanzadas como infraestructura como código (Terraform, Ansible) y plataformas de monitoreo centralizado aún no tienen sustento técnico ni roles definidos, lo que representa una viabilidad baja a corto plazo. Sin embargo, su incorporación paulatina permitiría completar el ciclo DevOps con observabilidad y gestión de entornos automatizada.

Luego de identificar las herramientas clave, se aplicó una matriz comparativa para evaluarlas según criterios como automatización, compatibilidad ágil, facilidad de uso, adopción en NGS y costos. Esta comparación permitió visualizar su viabilidad y facilitar la toma de decisiones técnicas alineadas al contexto organizacional.

La comparación se realizó mediante una valoración cualitativa basada en tres fuentes principales: evidencia documental de las herramientas, análisis técnico de sus funcionalidades, y su grado de uso actual en NGS según los resultados de los instrumentos aplicados (encuesta, Checklist técnico y FODA).

**Tabla 35** Evaluación Comparativa de Herramientas DevOps para Implementación CI/CD

Herramientas	Nivel de automatización	Integración con metodologías ágiles	Usabilidad intuitiva	Nivel de adopción actual en NGS	Costos de implementación	Observaciones
<b>Git / Azure DevOps Repos</b>	Alto	Alto	Medio-Alto	Sí	Bajo	Base sólida para versionamiento y control de tareas; ampliamente usado.
<b>Jenkins</b>	Alto	Medio	Medio	Parcial	Medio	Potente para CI/CD, pero requiere configuración técnica.
<b>Docker</b>	Medio	Medio	Medio	No	Medio	Necesita capacitación, clave para ambientes reproducibles.
<b>SonarQube</b>	Medio-Alto	Medio	Alto	No	Bajo-Medio	Mejora la calidad del código automáticamente, fácil integración.
<b>Terraform / Ansible (IaC)</b>	Alto	Bajo	Bajo-Medio	No	Medio-Alto	Potente para infraestructura automatizada; alta curva de aprendizaje.

Nota: Elaboración propia.

Esta matriz comparativa evidencia que herramientas como Git, Jenkins y Azure DevOps presentan un alto nivel de automatización y buena integración con metodologías ágiles, además de una adopción parcial en NGS, lo que facilita su consolidación como base del flujo CI/CD. En contraste, herramientas como Terraform o Prometheus, si bien ofrecen beneficios sustanciales en infraestructura como código y monitoreo avanzado, presentan barreras de entrada por su baja adopción y mayor complejidad técnica. Esta evaluación resalta la necesidad de priorizar herramientas con alta usabilidad y alineación operativa para lograr una implementación escalable y sostenible del enfoque DevOps.

Luego de la descripción de los resultados y análisis de las herramientas que pueden ayudar a que NGS implemente un flujo CD/CD integrado, presentamos los principales hallazgos sobre estas herramientas:

- Se cuenta con herramientas clave (Azure DevOps, Git), pero falta estandarización y

configuración adecuada.

- La integración de herramientas con métricas de calidad, despliegue y monitoreo es muy baja.
- La adopción de nuevas herramientas requiere un plan de formación y lineamientos de adopción organizacional.

Las herramientas identificadas en este estudio representan un componente fundamental para habilitar un flujo DevOps integrado y eficiente en NGS. Su adecuada selección y adopción permiten automatizar tareas críticas del ciclo de vida del software como la construcción, pruebas, despliegue y monitoreo. Aunque algunas ya están presentes en la organización, su uso no está completamente estandarizado ni alineado con las buenas prácticas. Por ello, es clave establecer lineamientos de adopción, formación técnica y una estrategia de integración gradual, con el fin de maximizar su valor y contribuir a un entorno de desarrollo más ágil, predecible y sostenible.

- **Políticas DevOps sugeridas para la implementación DevOps:** En una organización que busca adoptar DevOps de forma transversal, las políticas son fundamentales para trascender la implementación aislada de prácticas técnicas y alcanzar una institucionalización efectiva del enfoque. Mientras las herramientas y metodologías habilitan capacidades, son las políticas las que aseguran su aplicación coherente, sostenible y auditable a través de los proyectos y equipos. A continuación se muestran las políticas esenciales para lograr una estandarización robusta, confiable y replicable en un flujo CI/CD:

**Tabla 36** *Análisis de impacto y compatibilidad de políticas DevOps por fase del ciclo*

Política identificada	Si no se aplica	Si se aplica	Compatible con Ágil-DevOps-ITIL
Roles y responsabilidades formales (DevOps, QA, PO)	Ambigüedad en tareas, conflictos y cuellos de botella	Mayor coordinación, fluidez en planificación y ejecución	Roles claros en Scrum, DevOps y Dueños de proceso ITIL
Trazabilidad con versionamiento (Git)	Errores difíciles de auditar, regresiones sin control	Control de versiones, historial visible, rollback confiable	Trazabilidad ITIL y gestión de cambios alineada

Validaciones automáticas (linting, análisis estático)	Aumentan defectos en etapas siguientes	Mejora de calidad desde la fuente, mayor seguridad	DevOps técnico + enfoque en calidad ágil
Pruebas automatizadas como requisito	Pruebas manuales lentas, sin cobertura continua	Agilidad en la validación, reducción de bugs en producción	QA continuo y validación ITIL previa a entrega
Documentación mínima por entrega	Procesos frágiles, pérdida de conocimiento	Facilita onboarding, mejora consistencia y audibilidad	Gestión del conocimiento ITIL y acuerdos de definición de terminado en Scrum
Despliegue automatizado auditado	Errores de ambiente, falta de control	Estandarización, velocidad con confiabilidad	Automatización DevOps y control ITIL
Métricas y retroalimentación post-despliegue	Ceguera operativa, sin aprendizaje	Mejora continua basada en datos, reacción proactiva	Compatible con revisión de Sprint, mejora continua ITIL

Nota: Elaboración propia.

La ausencia de lineamientos formales en cada fase del ciclo DevOps ha llevado a una implementación fragmentada y variable entre equipos en NGS. Por ello, establecer políticas específicas permite transformar actividades técnicas individuales en un proceso unificado y repetible.

Luego de este análisis a nivel de políticas DevOps, se describen los siguientes hallazgos en esta dimensión:

**Tabla 37** Resumen de los hallazgos sobre políticas DevOps identificadas para NGS

Hallazgo	Impacto Sin Políticas DevOps	Beneficio Con Políticas DevOps
Ausencia de normativas de validación, código y despliegue	Fallos recurrentes, entregas manuales, baja trazabilidad	Entregas estables, automatizadas y trazables
Falta de política de monitoreo post-despliegue	No hay aprendizaje organizacional ni capacidad de respuesta rápida	Visibilidad operativa y mejora continua basada en métricas
Inexistencia de una gobernanza transversal del pipeline	Estándares y prácticas fragmentadas, sin alineación entre equipos	Coordinación interequipos, lineamientos unificados y sostenibilidad del modelo DevOps
No definición de roles como QA Automation o DevOps Engineer	Pruebas manuales, sin IaC, alto esfuerzo operativo	Roles técnicos especializados para automatización, calidad y eficiencia

No integración con marcos de trabajo existentes (Scrum, ITIL v4)	Incompatibilidad o implementación parcial de DevOps	Sinergia entre mejora continua, control de cambios y despliegue estable
--	---	---

Nota: Elaboración propia.

La alineación entre prácticas DevOps, herramientas tecnológicas y políticas normativas constituye un pilar fundamental para la institucionalización de un modelo operativo ágil, robusto y escalable en NGS. Integrar estas tres dimensiones no solo permite superar las brechas identificadas en el diagnóstico, sino que establece una base estandarizada y sostenible para la automatización de procesos, la mejora continua y la entrega de valor. La sinergia entre estas áreas garantiza que el ciclo CI/CD no se limite a esfuerzos aislados, sino que sea parte de una cultura organizacional coherente y transversal, alineada con marcos reconocidos como Scrum e ITIL V4.

**Figura 60** Sinergia entre prácticas, herramientas y políticas en el modelo DevOps



Nota: Elaboración propia.

#### 4.2.3 EVALUACIÓN DE LA ARQUITECTURA TECNOLÓGICA PARA LA IMPLEMENTACIÓN DEL FLUJO CI/CD

La evaluación de la arquitectura tecnológica constituye un elemento clave para determinar la viabilidad de la implementación de un flujo de Integración y Entrega Continua (CI/CD) en la consultora NGS. En este objetivo se analizan los componentes técnicos que sustentan los procesos de automatización, pruebas, despliegue, seguridad, escalabilidad, trazabilidad y operación de los sistemas, con el propósito de establecer en qué medida la infraestructura actual está preparada para soportar un modelo DevOps. Este análisis permite identificar fortalezas y limitaciones de la arquitectura existente, así como establecer una base objetiva para orientar las mejoras necesarias en función de los principios de automatización, confiabilidad, interoperabilidad y mejora continua.

A continuación, se muestran los instrumentos aplicados para el objetivo 3:

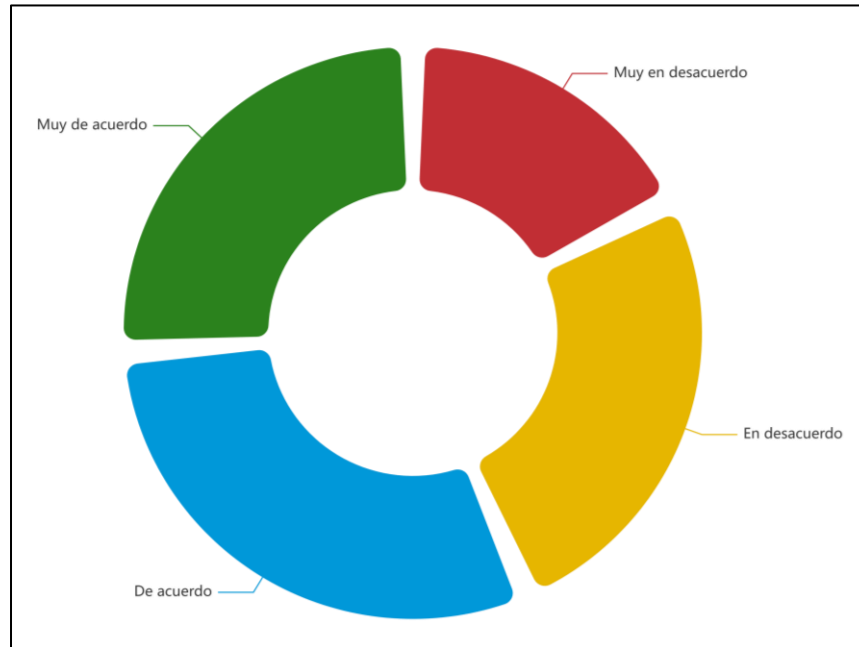
#### **ENCUESTA: AUTOMATIZACIÓN Y ARQUITECTURA TECNOLÓGICA**

En esta sección se evaluaron los componentes tecnológicos que permiten o limitan la implementación del flujo CI/CD en NGS. Los resultados indican que, aunque hay herramientas y ambientes que funcionan, la automatización no es completa, los entornos son diferentes y todavía hay procesos manuales que afectan la estabilidad y el seguimiento del ciclo de despliegue. La arquitectura actual evidencia un nivel de madurez intermedio, lo que indica la necesidad de mayor estandarización e integración entre sistemas.

A continuación, se describen los resultados obtenidos de las preguntas desarrolladas:

1. Los procesos de pruebas, despliegue y monitoreo se encuentran automatizados.

**Figura 61** Nivel de automatización en pruebas, despliegue y monitoreo

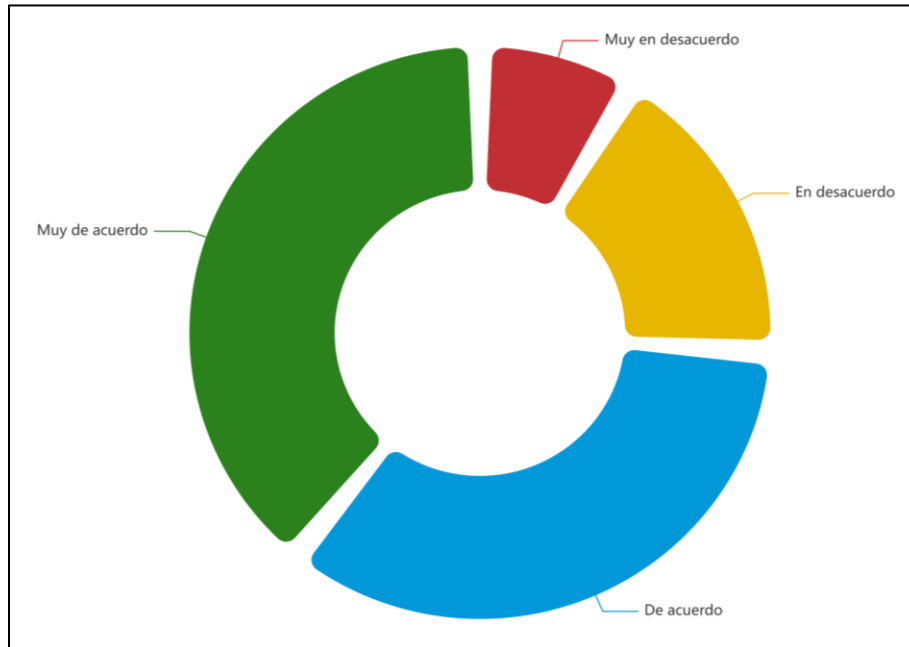


Nota: Elaboración propia.

El gráfico evidencia percepciones divididas respecto al nivel de automatización en los procesos de pruebas, despliegue y monitoreo dentro de NGS. Aunque existe un grupo significativo que considera que estos procesos sí presentan cierto grado de automatización, una proporción igualmente relevante expresa desacuerdo, lo que sugiere una implementación parcial, inconsistente y dependiente del equipo o proyecto. Esta variabilidad confirma los hallazgos generales de la investigación, donde se identificó que la automatización no es homogénea y que persisten actividades manuales que afectan la estabilidad y repetibilidad del flujo CI/CD.

2. La infraestructura tecnológica permite escalar los entornos (desarrollo, pruebas, producción) de forma ágil según la demanda.

**Figura 62** Capacidad de escalabilidad de los entornos tecnológicos en NGS

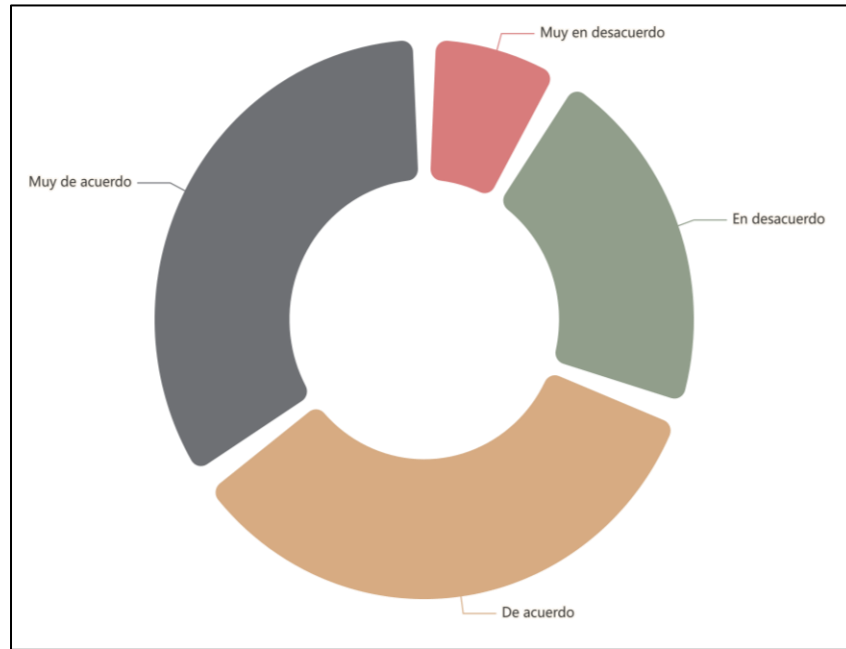


Nota: Elaboración propia.

El resultado muestra que la mayoría de los participantes considera que la infraestructura tecnológica de NGS sí permite escalar los entornos (desarrollo, pruebas y producción) de forma ágil según la demanda. Predominan las respuestas “De acuerdo” y “Muy de acuerdo”, lo que indica que existe una percepción positiva sobre la capacidad actual de la arquitectura para adaptarse a incrementos de carga o necesidades operativas específicas. No obstante, la presencia de respuestas en desacuerdo evidencia que esta capacidad no es homogénea y que algunos equipos o proyectos podrían experimentar limitaciones en la escalabilidad, ya sea por configuraciones manuales, falta de automatización del aprovisionamiento o variabilidad entre ambientes. En conjunto, el resultado sugiere un nivel de madurez intermedio: la infraestructura posee potencial para soportar un flujo CI/CD más dinámico, pero requiere estandarización y mejoras en automatización para garantizar que la escalabilidad sea consistente en toda la organización.

3. Existen controles de seguridad integrados en las etapas de integración y despliegue continuo.

**Figura 63** Integración de controles de seguridad en CI/CD

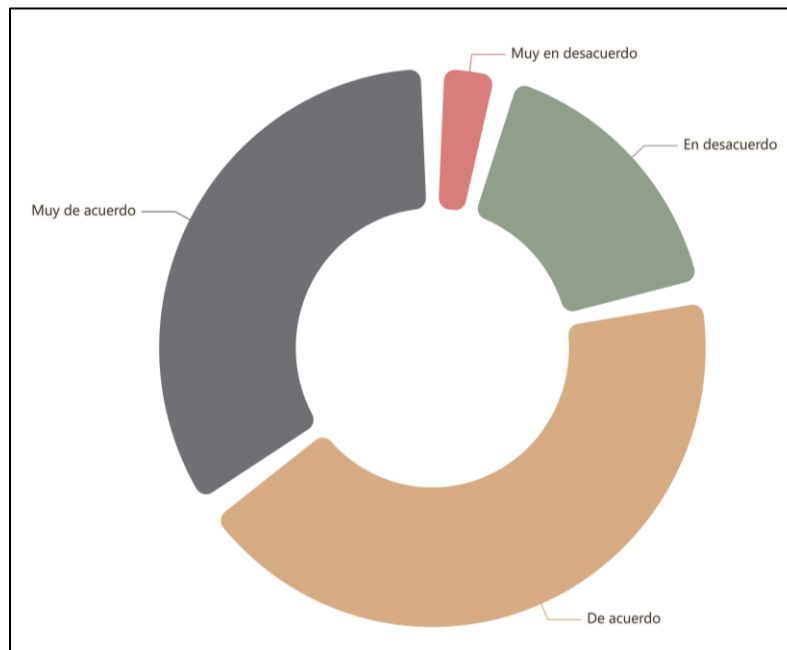


Nota: Elaboración propia.

La mayoría de los participantes considera que sí existen controles de seguridad integrados en las etapas de integración y despliegue continuo. Las respuestas “De acuerdo” y “Muy de acuerdo” predominan, lo que indica que NGS ha incorporado ciertos mecanismos de seguridad dentro del flujo técnico, aunque probablemente de manera parcial o concentrada en etapas específicas. Sin embargo, la presencia de respuestas en desacuerdo evidencia que estos controles no están plenamente estandarizados ni son aplicados de manera uniforme en todos los proyectos.

4. Cada despliegue cuenta con trazabilidad y registro (quién, cuándo y qué se modificó).

**Figura 64** Trazabilidad y registro de los despliegues



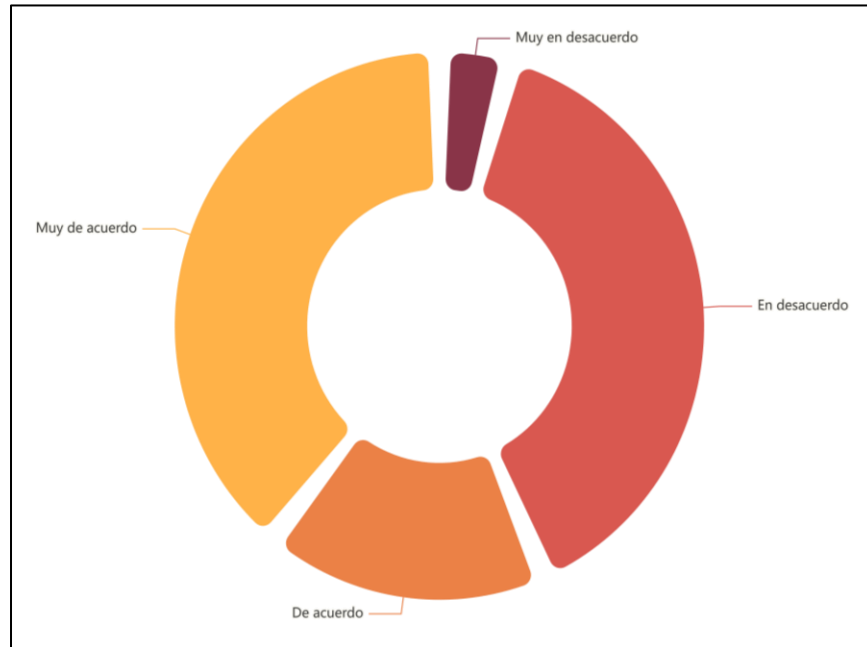
Nota: Elaboración propia.

Los resultados nos muestran que un gran número de los participantes percibe que los despliegues cuentan con trazabilidad y registro sobre quién realizó los cambios, cuándo se efectuaron y qué componentes fueron modificados. Las respuestas “De acuerdo” y “Muy de acuerdo” predominan, lo cual indica la existencia de mecanismos básicos de registro dentro del flujo actual. No obstante, la presencia de un grupo de participantes que manifiesta desacuerdo revela que esta trazabilidad no es uniforme ni consistentemente aplicada en todos los proyectos o ambientes.

Estos datos coinciden con los hallazgos del estudio, donde se identificó que la trazabilidad es una práctica presente, pero no estandarizada, y que persisten brechas en la integración entre cambios, incidentes, commits y despliegues. En conjunto, los resultados reflejan un nivel de madurez intermedio: existe avance en la adopción de controles de registro, pero se requiere fortalecer aún más.

5. La arquitectura actual contempla mecanismos de respaldo y recuperación ante fallos.

**Figura 65** *Mecanismos de respaldo y recuperación ante fallos*



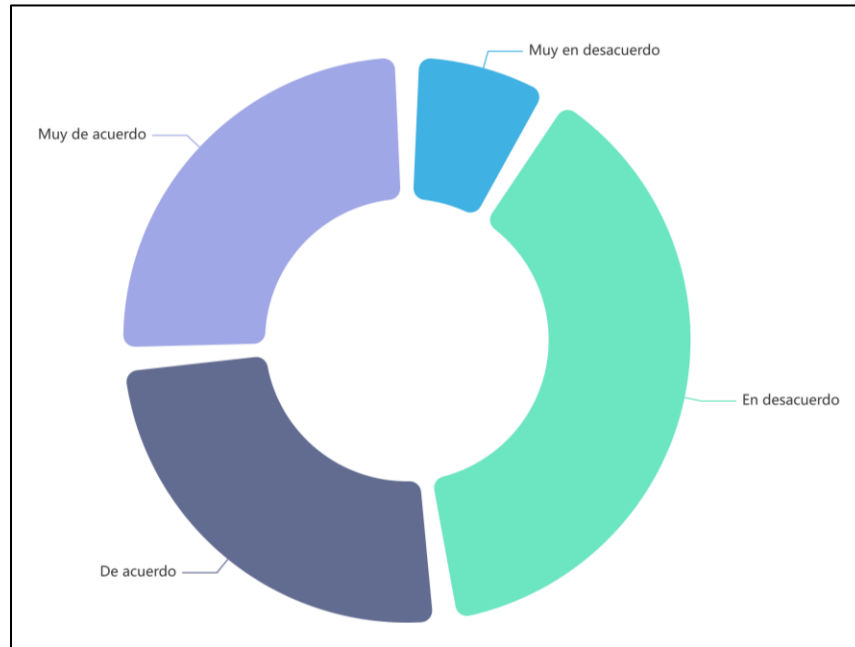
Nota: Elaboración propia.

El resultado de la encuesta nos muestra las participaciones divididas en los participantes respecto a si la arquitectura actual contempla mecanismos de respaldo y recuperación ante fallos. Aunque existe un porcentaje importante que afirma que sí se cuenta con estos mecanismos, especialmente quienes seleccionaron “Muy de acuerdo”, también destaca un grupo considerable que está en desacuerdo, lo que sugiere que estas capacidades no están implementadas de forma homogénea en todos los sistemas o ambientes.

Esta variabilidad confirma que, si bien NGS posee ciertos procedimientos de respaldo y recuperación, estos no están completamente estandarizados ni integrados dentro del flujo CI/CD. La falta de uniformidad puede generar riesgos en la estabilidad operativa y limitar la capacidad de responder de manera ágil ante incidentes, lo cual es fundamental para un entorno DevOps maduro.

6. La arquitectura actual de los sistemas está documentada y actualizada conforme a los cambios realizados en los proyectos.

**Figura 66** Actualización y documentación de la arquitectura tecnológica



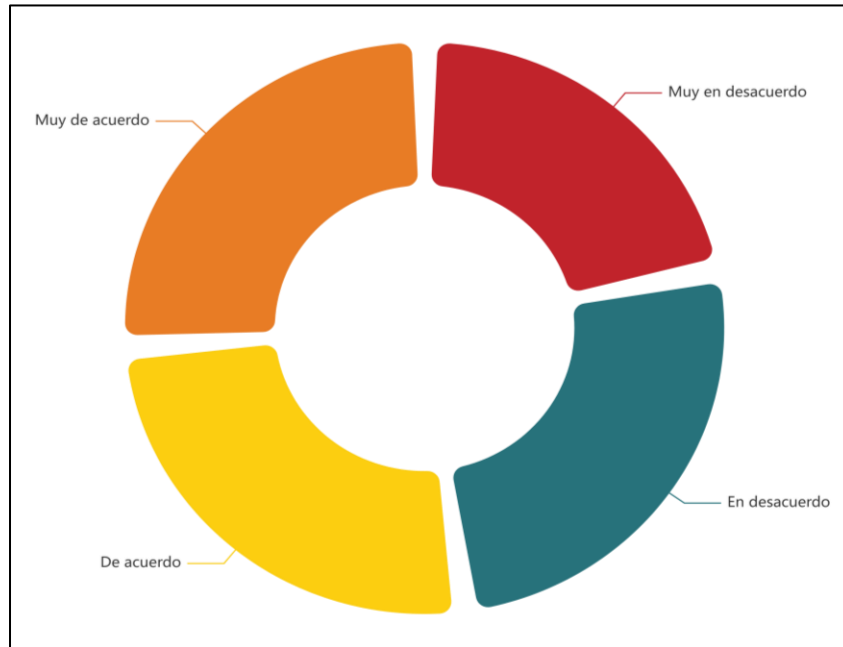
Nota: Elaboración propia.

El gráfico muestra una tendencia marcada hacia la percepción de que la arquitectura actual no está completamente documentada ni actualizada conforme a los cambios realizados en los proyectos. La categoría “En desacuerdo” es la más representativa, lo que indica que, para la mayoría de los participantes, la documentación arquitectónica es insuficiente, inconsistente o no se mantiene al ritmo de las modificaciones que ocurren durante el ciclo de desarrollo. Aunque un grupo importante señala estar “De acuerdo” o “Muy de acuerdo”, esta minoría sugiere que la actualización de la documentación depende en gran medida del equipo o proyecto, y no de un proceso institucional estandarizado.

Estos resultados coinciden con los hallazgos generales de la investigación, donde se identificó una brecha significativa en la formalización y mantenimiento de la arquitectura tecnológica, aspecto crítico para garantizar trazabilidad, estabilidad y coherencia en un flujo CI/CD. La falta de documentación actualizada dificulta la integración entre equipos, limita el análisis de impacto, afecta la gestión de cambios y reduce la capacidad de la organización para sostener un modelo DevOps

7. Existe un repositorio central de arquitectura o diagramas técnicos accesible para los equipos de desarrollo y operaciones.

**Figura 67** Disponibilidad de un repositorio central de arquitectura

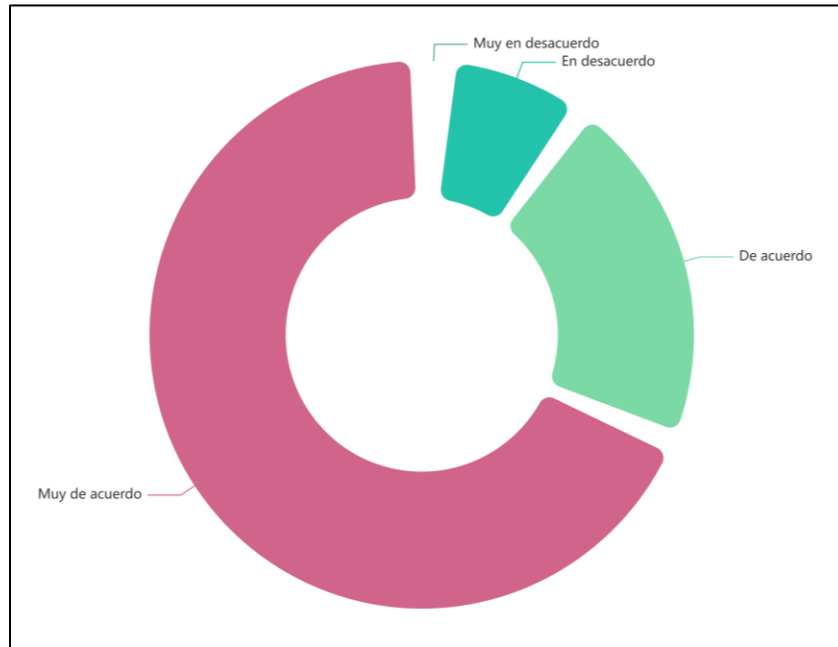


Nota: Elaboración propia.

El resultado evidencia que no existe una percepción unificada respecto a la presencia de un repositorio central de arquitectura accesible para los equipos de desarrollo y operaciones. Aunque una parte importante señala estar “De acuerdo” y “Muy de acuerdo”, indicando que sí hay algún tipo de repositorio, una proporción similar manifiesta estar en desacuerdo. Esto refleja que la centralización de la documentación técnica no es consistente ni conocida por todos los equipos. En el contexto de la tesis, este hallazgo confirma la necesidad de estandarizar la gestión documental y establecer un repositorio institucional único que facilite la colaboración y apoye la madurez del flujo CI/CD.

8. Se utilizan patrones o frameworks arquitectónicos definidos (por ejemplo, MVC, capas, servicios REST, SOA, etc.) en los proyectos de NGS.

**Figura 68** *Uso de patrones y frameworks arquitectónicos en NGS*



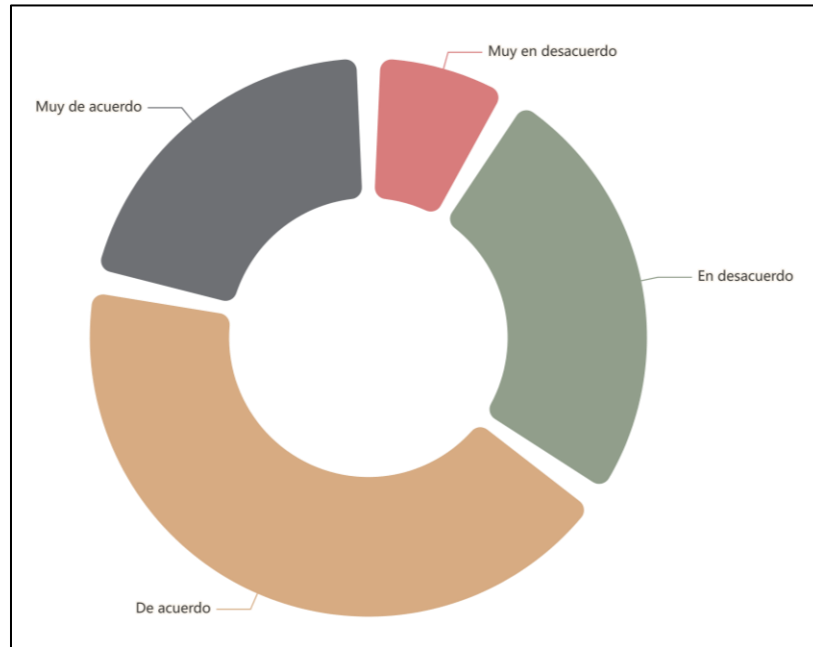
Nota: Elaboración propia.

El gráfico muestra una clara tendencia positiva respecto al uso de patrones y frameworks arquitectónicos definidos en los proyectos de NGS. La mayoría de los participantes seleccionó “Muy de acuerdo”, lo que indica que existe una adopción sólida de enfoques como MVC, arquitecturas por capas, servicios REST, SOA u otros modelos estructurados dentro del desarrollo de software. Esta percepción sugiere que la organización ha incorporado buenas prácticas de diseño y estructuración, lo cual favorece la mantenibilidad, escalabilidad y estandarización de los proyectos.

Las respuestas minoritarias en desacuerdo demuestran que, aunque la mayoría aplica estos patrones, aún podría haber equipos que no utilizan de forma uniforme los lineamientos arquitectónicos establecidos o que emplean variantes no estandarizadas.

9. Se realizan evaluaciones técnicas o revisiones arquitectónicas antes de implementar nuevas soluciones o componentes

**Figura 69** Realización de revisiones técnicas y evaluaciones arquitectónicas



Nota: Elaboración propia.

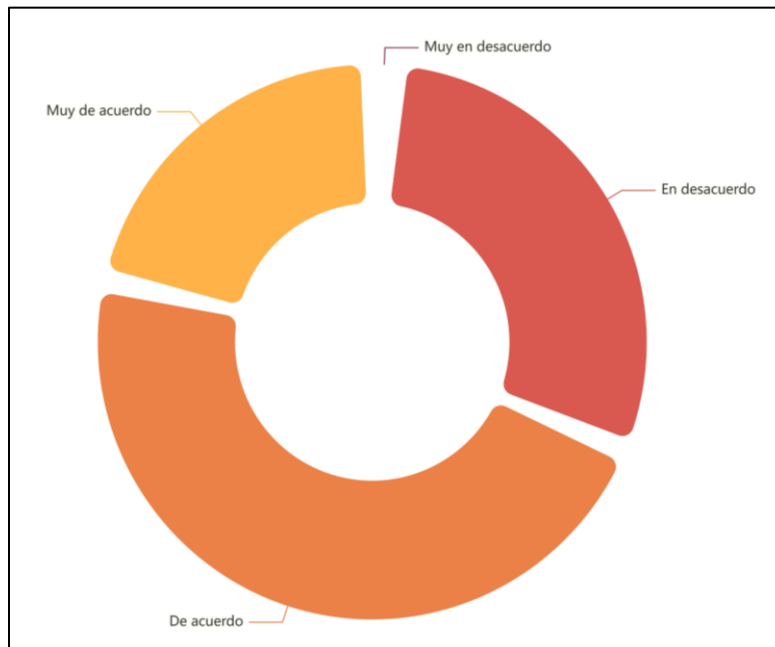
Podemos ver que el resultado indica que la mayoría de los participantes percibe que sí se realizan evaluaciones técnicas o revisiones arquitectónicas antes de implementar nuevas soluciones o componentes, predominando las respuestas “De acuerdo”. Esto sugiere que NGS cuenta con prácticas básicas de validación técnica previas al desarrollo o despliegue, lo cual es positivo para garantizar coherencia y reducir riesgos en los proyectos.

No obstante, la presencia significativa de respuestas en desacuerdo revela que estas revisiones no son aplicadas de manera uniforme en todos los equipos o iniciativas. La existencia de criterios y prácticas heterogéneas coincide con los hallazgos generales del estudio, que muestran falta de estandarización en procesos y documentación arquitectónica. La menor proporción de participantes “Muy de acuerdo” refuerza la idea de que estas evaluaciones existen, pero no alcanzan aún un nivel de madurez institucional ni se integran plenamente al flujo CI/CD. el resultado refleja un nivel de madurez intermedio: hay conciencia y aplicación de revisiones técnicas, pero la organización requiere formalizar, estandarizar y documentar estos procesos para fortalecer la calidad, reducir retrabajos y garantizar una toma de decisiones arquitectónicas coherente y alineada con un modelo DevOps más disciplinado.

10. La arquitectura actual facilita la integración de herramientas de monitoreo, despliegue

continuo y control de versiones.

**Figura 70** *Facilidad para integrar herramientas de monitoreo, CI/CD y control de versiones*

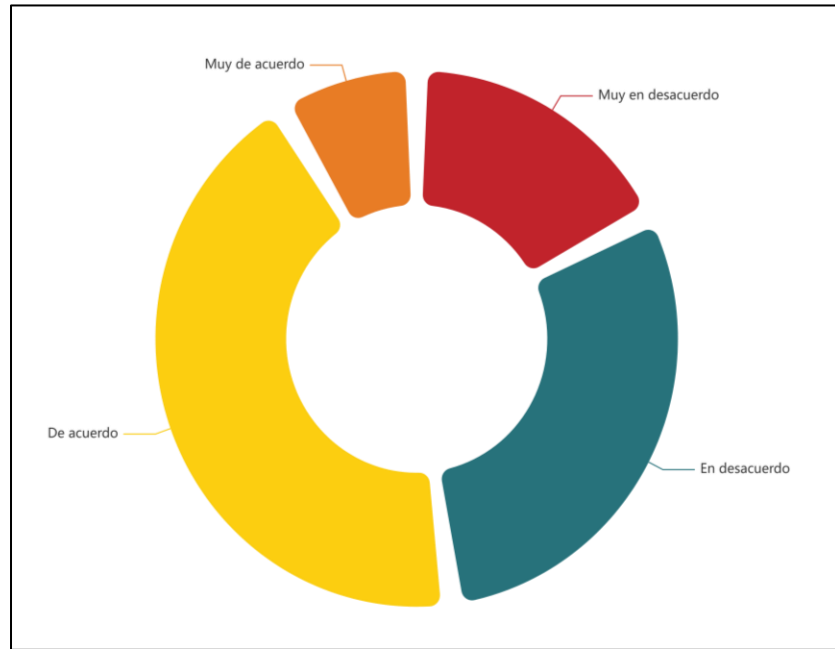


Nota: Elaboración propia.

El gráfico evidencia que la mayoría de los participantes considera que la arquitectura actual de NGS sí facilita la integración de herramientas de monitoreo, despliegue continuo y control de versiones, ya que la opción “De acuerdo” presenta la mayor proporción de respuestas. Esto sugiere que la organización cuenta con una base tecnológica que permite conectar diferentes componentes del flujo CI/CD, lo cual es positivo para la automatización y la trazabilidad del ciclo de entrega; sin embargo, la presencia significativa de respuestas en desacuerdo indica que esta integración no es homogénea en todos los proyectos o equipos. Esto coincide con los hallazgos generales de la investigación, que señalan la existencia de configuraciones aisladas, falta de estandarización y brechas en la integración completa entre herramientas clave como monitoreo, pipelines y repositorios de versiones.

11. Los servicios y componentes críticos cuentan con mecanismos de redundancia o balanceo de carga para garantizar su disponibilidad.

**Figura 71** Disponibilidad y resiliencia de servicios críticos

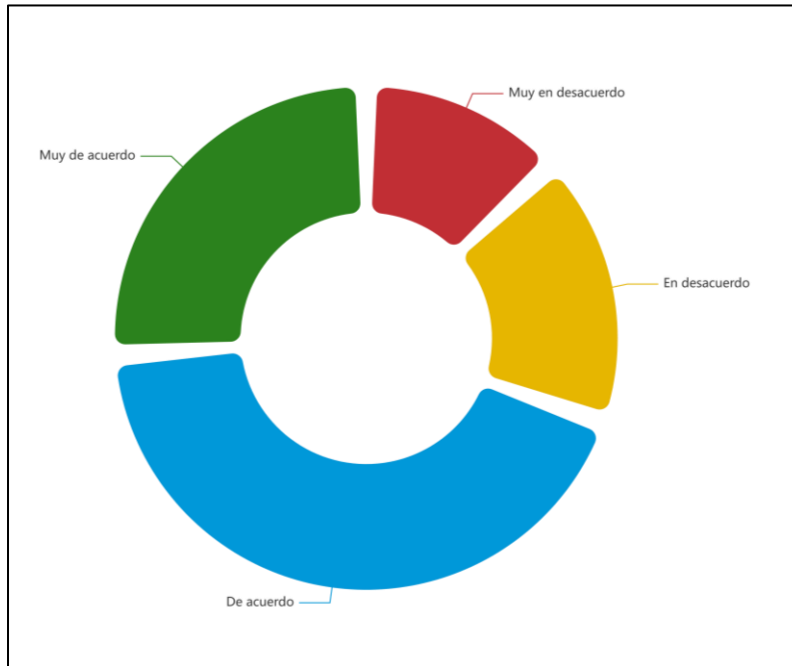


Nota: Elaboración propia.

La mayoría de los participantes considera que los servicios y componentes críticos sí cuentan con mecanismos de redundancia o balanceo de carga para garantizar su disponibilidad, predominando las respuestas “De acuerdo”. Esto indica que NGS ha implementado algunas prácticas de alta disponibilidad que permiten sostener la continuidad operativa ante fallos o incrementos de demanda, pero el número notable de respuestas en desacuerdo evidencia que estas capacidades no están completamente estandarizadas o no son uniformes en todos los sistemas. Esta percepción coincide con los hallazgos generales de la investigación, que señalan brechas en la consistencia de la arquitectura tecnológica y la necesidad de fortalecer los mecanismos de resiliencia y recuperación. La baja presencia de respuestas “Muy de acuerdo” sugiere que, aunque existen mecanismos de disponibilidad, estos podrían ser limitados o estar implementados parcialmente.

12. Se utilizan contenedores o servicios en la nube para facilitar la escalabilidad del flujo CI/CD.

**Figura 72** *Uso de contenedores o servicios en la nube para escalar CI/CD*

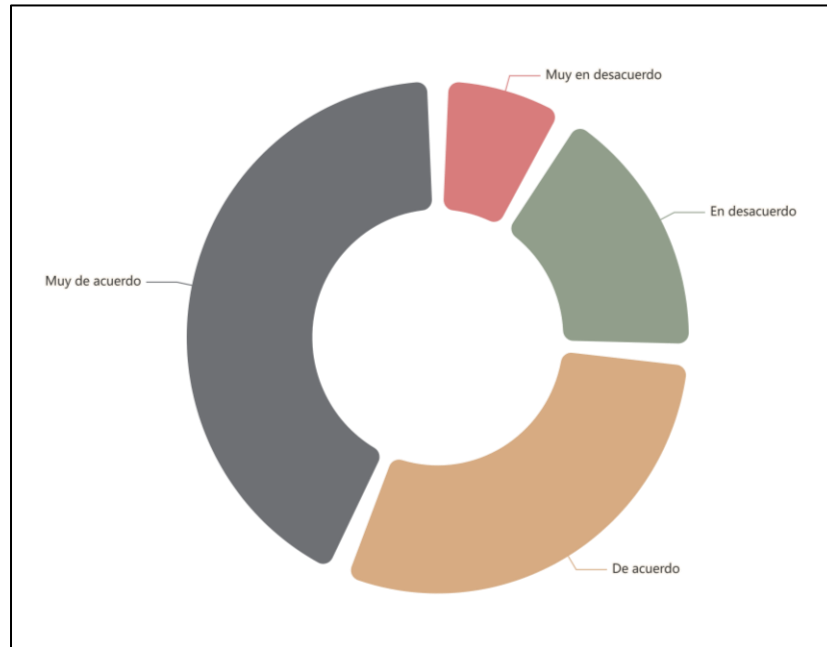


Nota: Elaboración propia.

El resultado de las respuestas muestra que la mayoría de los participantes está “De acuerdo” en que NGS utiliza contenedores o servicios en la nube para facilitar la escalabilidad del flujo CI/CD. Esto indica que la organización ha comenzado a adoptar tecnologías modernas como contenedores, virtualización ligera o servicios cloud que permiten flexibilizar y escalar los entornos de desarrollo, pruebas y despliegue, pero se muestra también la presencia visible de respuestas en desacuerdo; demuestra que este uso no es uniforme ni completamente adoptado por todos los equipos o proyectos. Esta variabilidad coincide con los hallazgos generales de la investigación, que señalan la existencia de prácticas dispersas y niveles de adopción diferentes en la arquitectura tecnológica. La menor proporción de respuestas “Muy de acuerdo” refleja que, aunque existen iniciativas del llenado o uso de servicios en la nube, estas aún no se han consolidado como parte estructural del flujo CI/CD.

13. Las credenciales y accesos son gestionados mediante herramientas seguras.

**Figura 73** Gestión segura de credenciales y accesos



Nota: Elaboración propia.

Los participantes, en su mayoría consideran que la gestión de credenciales y accesos se realiza mediante herramientas seguras, predominando las respuestas “De acuerdo” y “Muy de acuerdo”. Esto indica que existen prácticas de seguridad relativamente consolidadas dentro de los procesos de NGS. Sin embargo, la presencia de respuestas en desacuerdo evidencia que estas herramientas no se aplican de manera uniforme en todos los equipos o proyectos.

14. Responda abiertamente: Desde su experiencia, ¿Qué mejoras considera necesarias en la arquitectura tecnológica para aumentar la confiabilidad, escalabilidad o interoperabilidad de los sistemas?

**Tabla 38** Descripción y análisis de las respuestas abiertas.

Aspecto clave	Descripción breve	Análisis
<b>Modernización y escalabilidad de la arquitectura</b>	Se propone actualizar infraestructura, migrar a microservicios, usar contenedores y escalar componentes según demanda.	La modernización mediante microservicios, contenedores y escalabilidad granular incrementa confiabilidad, flexibilidad y rendimiento. Permite escalar partes específicas del sistema, optimizando costos y desempeño en entornos CI/CD.

<b>Integración y estandarización técnica</b>	Se sugiere estandarizar integraciones, usar APIs documentadas (REST, GraphQL), buses ESB o mensajería (RabbitMQ, Kafka).	La estandarización mejora la interoperabilidad y comunicación entre sistemas heterogéneos. Es clave para arquitecturas modulares y para facilitar la integración con clientes o plataformas distintas.
<b>Observabilidad y monitoreo</b>	Se recomienda incorporar monitoreo centralizado, logging estructurado y alertas proactivas.	La observabilidad robusta permite detectar fallos temprano y mantener estabilidad en CI/CD. Es esencial para DevOps e ITIL V4, habilitando decisiones informadas y respuesta oportuna a incidentes.
<b>Automatización y pruebas continuas</b>	Se destaca facilitar pruebas automatizadas de integración y realizar despliegues sin afectar todo el sistema.	Automatizar pruebas y aplicar técnicas como blue-green o canary releases reduce riesgos, mejora calidad y permite cambios seguros sin interrumpir el sistema completo.
<b>Documentación, gobernanza y capacitación</b>	Se menciona mejorar documentación de procesos, capacitar al personal y definir un framework interno de trabajo.	La gobernanza arquitectónica sólida mejora consistencia y trazabilidad. La documentación y capacitaciones aseguran que las prácticas se apliquen de forma uniforme y reducen la dependencia del conocimiento tácito.
<b>Aislamiento de fallos y robustez del sistema</b>	Se propone aislar fallos por servicio y contar con una arquitectura robusta.	Aislar fallos mejora resiliencia y disponibilidad del sistema. Una arquitectura robusta soporta mejor crecimiento, escalabilidad y operaciones continuas.
<b>Pruebas desde el nivel del desarrollador (shift-left)</b>	Se solicita incorporar pruebas desde etapas tempranas del desarrollo.	El enfoque shift-left mejora la calidad desde el inicio, reduce el retrabajo y acelera el ciclo CI/CD, alineándose con las mejores prácticas DevOps.

Nota: Elaboración propia.

A continuación, se presenta un resumen consolidado de los resultados obtenidos en la sección de la encuesta relacionada con automatización y arquitectura tecnológica. Esta tabla sintetiza las principales dimensiones evaluadas, sus resultados más representativos y una interpretación técnica inicial que servirá como base para el análisis detallado en los apartados siguientes.

**Tabla 39** Resumen de los resultados de las preguntas de la encuesta

<b>Dimensión Evaluada</b>	<b>Resultado Principal</b>	<b>Interpretación Técnica</b>
<b>Automatización CI/CD</b>	Parcial y no estandarizada	Existen pipelines, pero no son homogéneos entre proyectos. Aún hay dependencias manuales.
<b>Escalabilidad</b>	Medianamente aceptable	Algunos entornos permiten escalar, pero no todos los proyectos lo logran.
<b>Contenedores y nube</b>	Uso inicial	Se aplican en proyectos puntuales, no como estándar institucional.
<b>Seguridad en CI/CD</b>	Aceptable	Se aplican controles de acceso, credenciales y revisiones básicas.
<b>Trazabilidad de despliegues</b>	Moderada	Hay registros de cambios, pero no siempre centralizados.
<b>Alta disponibilidad</b>	Parcial	Existen mecanismos técnicos, pero no están estandarizados en todos los sistemas.
<b>Respaldo y recuperación</b>	Inconsistente	Hay backups, pero no integrados completamente al pipeline CI/CD.
<b>Documentación</b>	Deficiente	Los cambios no siempre están documentados ni actualizados.
<b>Repositorio central de arquitectura</b>	Parcial	No todos los equipos conocen o utilizan un repositorio único.
<b>Patrones arquitectónicos</b>	Fortalecido	Se aplican buenas prácticas como MVC, capas y servicios.
<b>Monitoreo y observabilidad</b>	Limitado	Falta integración completa de logs, métricas y alertas proactivas.

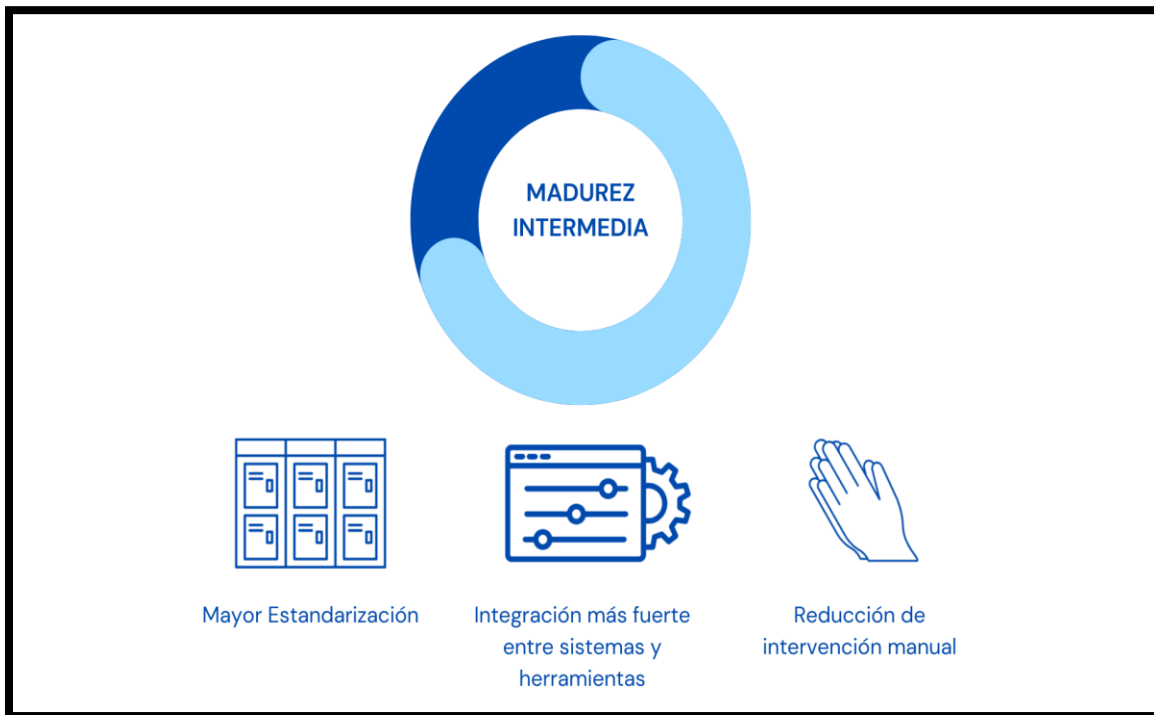
Nota: Elaboración propia

Con base en los resultados obtenidos en la sección de Automatización y Arquitectura Tecnológica de la encuesta, se procederá a realizar un análisis más profundo de la situación actual de NGS. Este análisis se centrará en tres ejes clave para la implementación de un flujo CI/CD sostenible: escalabilidad, seguridad y trazabilidad. Cada uno de estos aspectos será evaluado en función de la información reportada por los equipos, permitiendo identificar limitaciones técnicas y oportunidades de mejora dentro de la arquitectura tecnológica existente.

- **Evaluación de la Arquitectura Tecnológica Actual:** La arquitectura tecnológica evidencia que NGS presenta una madurez intermedia, coherente con las brechas detectadas en procesos, herramientas y competencias. La automatización parcial, la falta de estandarización, la débil documentación y la limitada observabilidad confirman que las debilidades no son solo organizativas, sino también técnicas. La adopción incipiente de contenedores, servicios en la nube y alta disponibilidad demuestra que existe capacidad tecnológica instalada, aunque aún no consolidada, se confirma que la arquitectura es compatible con las prácticas DevOps y con los procesos de ITIL V4 en gestión de cambios, despliegues, seguridad e incidentes. No obstante, la fragmentación del pipeline CI/CD limita su aplicación a nivel institucional. En conjunto, los resultados validan la necesidad de fortalecer el gobierno arquitectónico, estandarizar la automatización y robustecer la observabilidad para asegurar una entrega continua sostenible.

La arquitectura en NGS se ubica en un nivel de madurez intermedio, con necesidad clara de:

**Figura 74** Nivel de madurez de la arquitectura en NGS



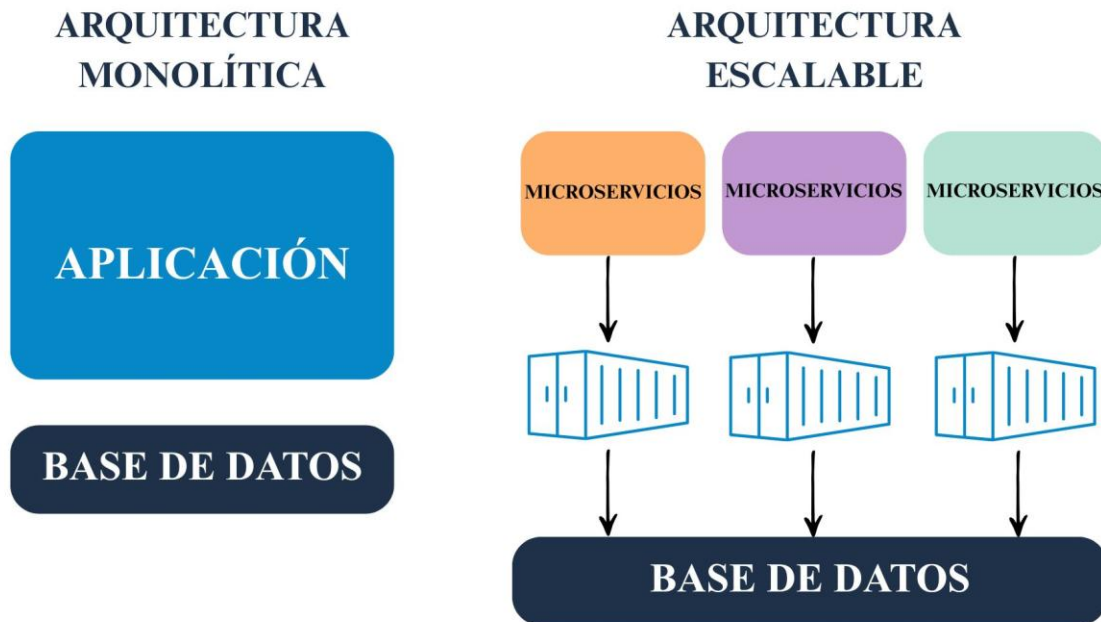
Nota: Elaboración propia.

- **Escalabilidad:** La arquitectura tecnológica de NGS presenta una capacidad de escalabilidad limitada y heterogénea. Según los resultados obtenidos, si bien algunos proyectos han logrado integrar servicios sobre entornos parcialmente escalables, esta práctica no está institucionalizada ni respaldada por mecanismos técnicos homogéneos. El uso de tecnologías como contenedores o infraestructura en la nube es aún incipiente y restringido a iniciativas específicas, lo que dificulta su adopción transversal. Esto implica que la incorporación de nuevos servicios o pipelines puede comprometer el rendimiento si no se acompaña de una arquitectura flexible y preparada para la expansión horizontal.

Actualmente, NGS no cuenta con una infraestructura dinámica que permita la creación o modificación de entornos de manera automatizada. Esta limitación reduce la capacidad para escalar procesos críticos del flujo CI/CD de forma ágil y segura. La falta de una estrategia de estandarización en el uso de cloud y contenedores restringe también la reutilización de entornos y la portabilidad de las soluciones. En este contexto, escalar sin degradar el rendimiento resulta complejo, especialmente ante demandas simultáneas o crecimiento acelerado de proyectos.

A continuación se muestra visualmente la diferencia entre una arquitectura monolítica y una escalable basada en microservicios y contenedores, en el contexto del análisis de escalabilidad en NGS. Este contraste permite evidenciar que, mientras la arquitectura actual aún mantiene elementos rígidos y poco desacoplados, la proyección hacia una arquitectura basada en contenedores y despliegue dinámico permitiría mayor flexibilidad, facilidad de crecimiento por componentes, y soporte para múltiples pipelines paralelos sin degradar el rendimiento. Esta transición resulta clave para soportar de forma sostenible un flujo CI/CD completo y alineado con prácticas modernas de DevOps.

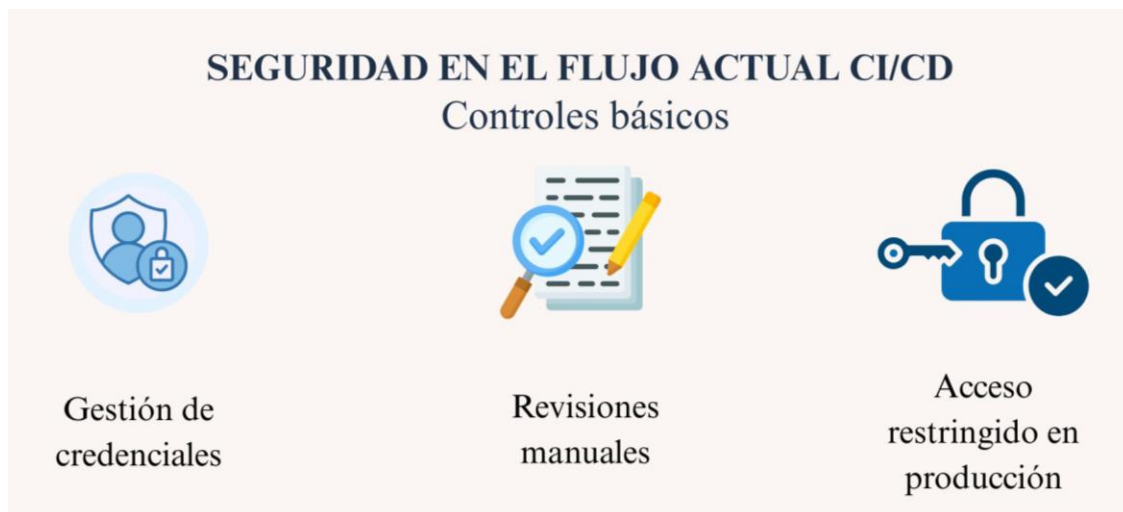
**Figura 75** Comparativa entre arquitectura monolítica vs. escalable con contenedores y microservicios



Nota: Elaboración propia.

- **Seguridad:** La seguridad dentro del flujo CI/CD de NGS se encuentra en un estado aceptable, aunque aún con notables áreas de mejora. Los resultados indican que se aplican controles básicos como gestión de credenciales, revisiones manuales y acceso restringido a ambientes productivos. Sin embargo, se evidencia la falta de una integración plena de seguridad dentro del pipeline, lo que implica que los mecanismos actuales no están automatizados ni presentes en todas las etapas del desarrollo y despliegue.

**Figura 76** Seguridad en el flujo actual CI/CD en NGS



Nota: Elaboración propia.

- **Trazabilidad:** La trazabilidad en el flujo CI/CD de NGS presenta avances parciales, pero aún se encuentra lejos de un nivel óptimo que garantice control y auditabilidad en todas las etapas del ciclo de desarrollo. Los hallazgos de la encuesta indican que si bien existen registros de cambios y versiones, estos no están centralizados ni normalizados, lo cual dificulta reconstruir el historial completo de una entrega o diagnosticar fallos en producción. Este tipo de trazabilidad fragmentada impide una visibilidad de extremo a extremo y puede comprometer tanto la calidad del software como la capacidad de respuesta ante incidentes.

**Figura 77** Trazabilidad en el flujo de actual de CI/CD en NGS



Nota: Elaboración propia.

Para contrastar esta situación y definir lineamientos de mejora, se llevó a cabo una revisión documental especializada, orientada a identificar modelos de arquitectura tecnológica comúnmente recomendados para entornos CI/CD bajo principios DevOps. Se sugiere que una arquitectura moderna, sostenible y alineada con estos principios se fundamenta en un enfoque modular y automatizado.

La documentación técnica de proveedores líderes como Amazon Web Services (AWS, 2023) y Microsoft Azure (2024) complementa este enfoque, al describir arquitecturas de referencia que integran componentes clave como los siguientes:

**Figura 78** Componentes clave de una arquitectura tecnológica moderna para flujos CI/CD bajo enfoque DevOps



Nota: Elaboración propia.

Estos elementos ofrecen una base sólida para habilitar despliegues continuos, automatizados, trazables y seguros.

Luego de este análisis realizado para el objetivo 3, se muestran los principales hallazgos encontrados:

**Figura 79** Principales hallazgos arquitectónicos del flujo CI/CD en NGS



Nota: Elaboración propia.

#### 4.2.4 INDICADORES DE DESEMPEÑO Y SU IMPACTO EN LOS TIEMPOS ENTREGA CONTINUA DEL SERVICIO

La medición del desempeño es un componente fundamental para garantizar los tiempos entrega continua de los servicios de TI en un entorno orientado a DevOps. En este objetivo se analizan los indicadores clave de desempeño (KPIs) como herramientas estratégicas para evaluar la eficiencia de los procesos, la calidad del software, la estabilidad de los despliegues y la satisfacción del cliente. La correcta definición y utilización de estos indicadores permite a la organización tomar decisiones basadas en datos, identificar oportunidades de optimización y asegurar la alineación entre los objetivos tecnológicos y los objetivos del negocio. En este contexto, la evaluación de los KPIs en NGS se orienta a determinar en qué medida la organización cuenta con mecanismos formales de seguimiento y control que respalden un modelo de entrega continua del servicio.

A continuación, se muestran los instrumentos aplicados para el objetivo 4:

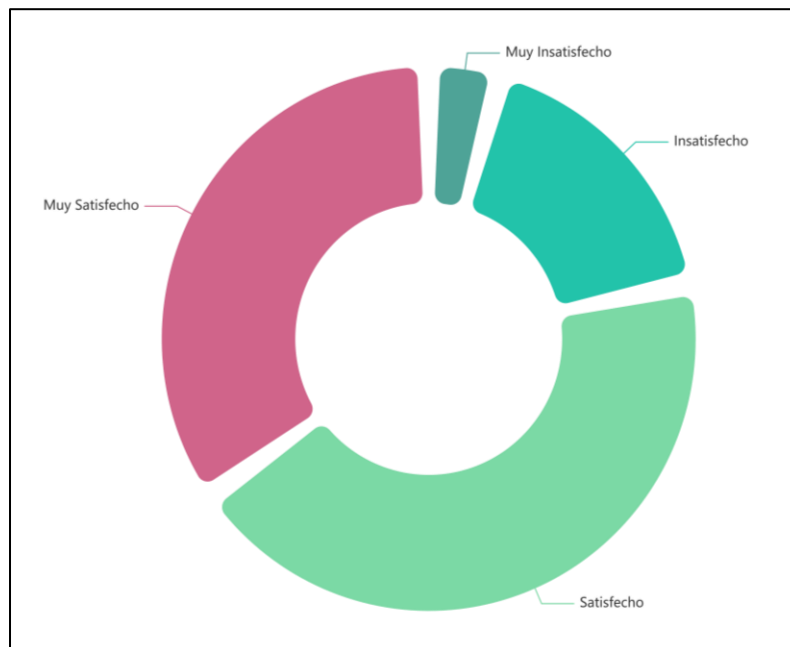
## ENCUESTA: KPIs E IMPACTO EN ENTREGA CONTINUA:

En esta sección se evidencia que, si bien existe conciencia sobre la importancia de medir el desempeño mediante indicadores, la implementación de KPIs en NGS aún no se encuentra plenamente formalizada ni sistematizada. Las métricas relacionadas con tiempos de entrega, calidad del software, frecuencia de despliegues y atención de incidencias se aplican de forma parcial, en muchos casos sin un esquema de medición estandarizado ni tableros de control consolidados. Los resultados indican que NGS cuenta con oportunidades significativas para fortalecer la gestión de su desempeño mediante la adopción estructurada de KPIs, lo que permitiría respaldar la toma de decisiones, optimizar la entrega continua y reforzar la mejora continua del servicio.

A continuación, se presentan y describen las preguntas y respuestas desarrolladas:

1. ¿Qué tanto ha mejorado el tiempo promedio de entrega de versiones en los últimos meses?

**Figura 80** Mejora percibida en el tiempo promedio de entrega de versiones durante los últimos meses.



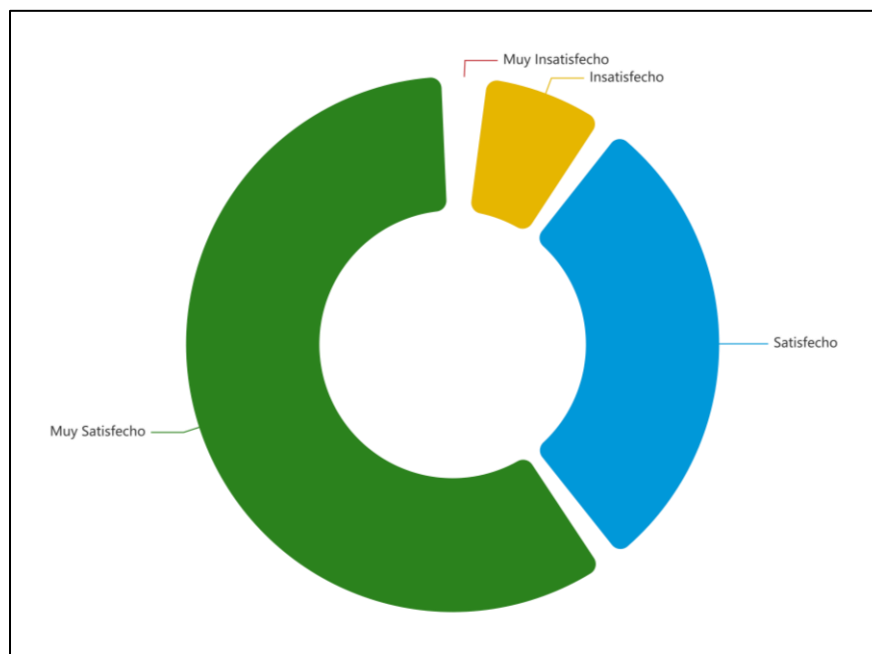
Nota: Elaboración propia.

Los resultados muestran mayoritariamente positiva respecto a la mejora en el tiempo promedio de entrega de versiones en NGS, predominando las valoraciones de “Satisfecho” y “Muy satisfecho”. Esto indica que los participantes reconocen avances en la rapidez con la que las nuevas

versiones de software están siendo liberadas, lo cual sugiere un fortalecimiento progresivo de los procesos de desarrollo y despliegue. La presencia de algunas valoraciones de insatisfacción es reducida, lo que refleja que los casos donde no se percibe mejora son puntuales. En conjunto, este resultado permite concluir que NGS ha logrado avances favorables en la optimización de sus tiempos de entrega, lo que impacta positivamente en la entrega continua; no obstante, aún existe margen para seguir perfeccionando la automatización y la estandarización del flujo CI/CD.

2. ¿Qué tan estable considera la calidad del software desplegado?

**Figura 81** *Percepción de estabilidad en la calidad del software desplegado.*



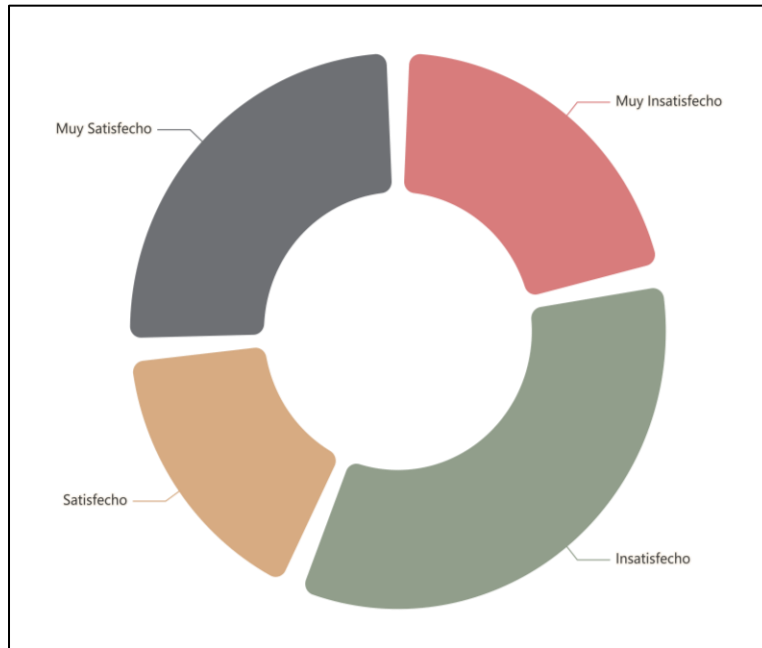
Nota: Elaboración propia.

Las respuestas muestran una mayoría positiva respecto a la estabilidad del software desplegado por NGS, predominando las valoraciones de “Satisfecho” y “Muy satisfecho”. Esto indica que, en términos generales, los participantes consideran que las versiones liberadas mantienen un comportamiento estable, con un nivel aceptable de fallos en producción. La presencia de valoraciones de insatisfacción es mínima, lo que sugiere que los problemas de inestabilidad se presentan de forma aislada y no recurrente. En conjunto, los resultados permiten concluir que NGS presenta un buen nivel de estabilidad en la calidad del software desplegado, lo que fortalece la confianza en el proceso de entrega continua; sin embargo, se mantiene la oportunidad de seguir reforzando las pruebas automatizadas y los controles de calidad para

consolidar aún más esta percepción.

3. ¿Qué tan conocidos son los indicadores clave de desempeño por parte del equipo técnico?

**Figura 82** Nivel de conocimiento de los indicadores clave de desempeño (KPIs) por parte del equipo técnico.

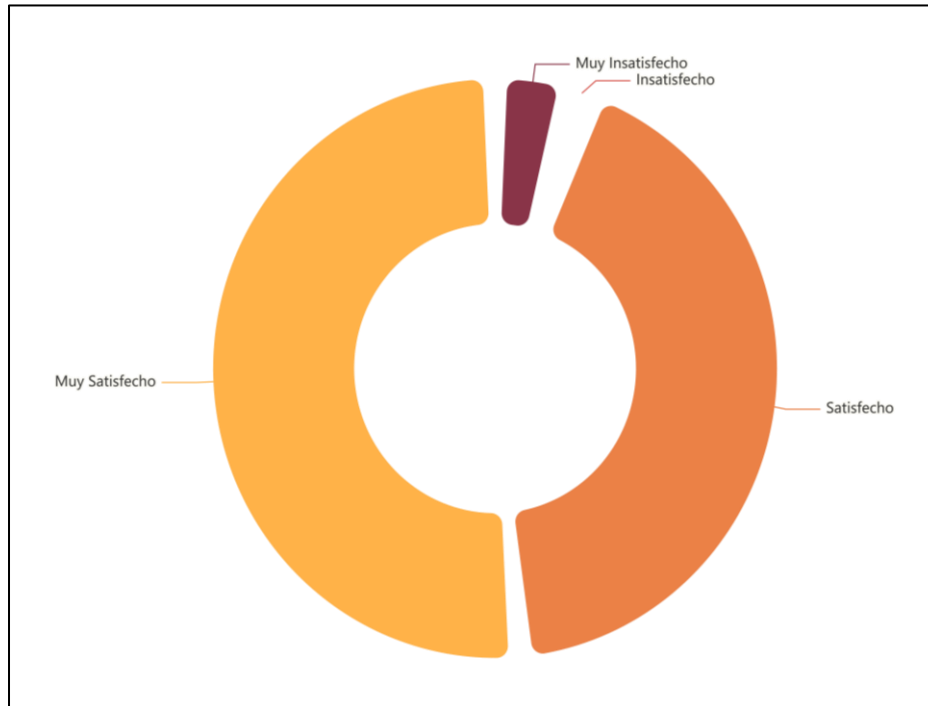


Nota: Elaboración propia.

Las respuestas muestran resultado diverso respecto al nivel de conocimiento de los indicadores clave de desempeño (KPIs) por parte del equipo técnico de NGS, ya que se presentan valoraciones distribuidas entre satisfacción, insatisfacción y muy satisfecho. Esta variabilidad indica que, aunque una parte del equipo reconoce y comprende los KPIs, otra proporción significativa aún no los maneja de manera clara o completamente. La presencia de valoraciones de insatisfacción refleja que el conocimiento sobre estos indicadores no está plenamente estandarizado ni difundido en toda la organización. En conjunto, los resultados permiten concluir que NGS presenta un nivel intermedio de madurez en la gestión del conocimiento de KPIs, por lo que se vuelve necesario fortalecer la capacitación, la comunicación interna y la institucionalización de estas métricas como herramientas estratégicas para apoyar la mejora continua y la toma de decisiones basadas en datos.

4. Los clientes están satisfechos con los entregables.

**Figura 83** Grado de satisfacción de los clientes con los entregables de software.

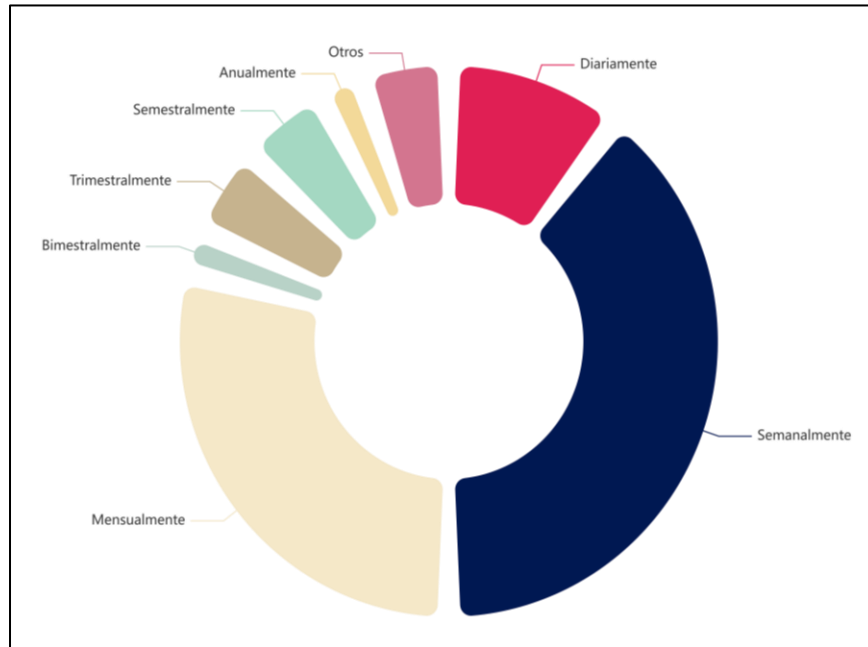


Nota: Elaboración propia.

Los resultados muestran mayoría positiva sobre el nivel de satisfacción de los clientes con los entregables proporcionados por NGS, ya que predominan claramente las valoraciones de “Satisfecho” y “Muy satisfecho”. Esto indica que los productos entregados cumplen, en general, con las expectativas de los clientes en términos de funcionalidad, calidad y cumplimiento de requisitos. La presencia de valoraciones de insatisfacción es mínima, lo que sugiere que los casos de inconformidad son aislados y no representan una tendencia general. En conjunto, estos resultados permiten concluir que NGS mantiene un alto nivel de aceptación en sus entregables, lo que fortalece su credibilidad como proveedor de servicios de software; no obstante, se reconoce la importancia de sostener procesos de mejora continua para conservar y elevar este nivel de satisfacción.

5. ¿Con qué frecuencia se realizan despliegues o entregas de software en NGS?

**Figura 84** Frecuencia de despliegues o entregas de software en NGS.



Nota: Elaboración propia.

Las respuestas muestran que la frecuencia de entregas de software en NGS se concentra principalmente en periodos semanales y mensuales, lo que indica que la organización mantiene un ritmo de liberaciones relativamente constante. También se observan frecuencias menores en periodos trimestrales, semestrales y casos aislados diarios, lo que refleja que la periodicidad de los despliegues varía según el tipo de proyecto, cliente o nivel de complejidad. Esta dispersión evidencia que no existe aún un patrón completamente estandarizado de entregas. En conclusión, aunque NGS demuestra capacidad para realizar entregas de forma periódica, aún presenta un nivel de madurez intermedio en la gestión de la frecuencia de despliegues, por lo que la adopción más formal de CI/CD permitiría avanzar hacia entregas más frecuentes, predecibles y automatizadas.

6. Responda abiertamente: Desde su experiencia, ¿Qué indicadores considera más útiles para medir la eficiencia del proceso de entrega continua?

**Tabla 40** Indicadores considerados más útiles para medir la eficiencia del proceso de entrega continua según la experiencia del personal encuestado

Aspecto clave	Descripción breve	Análisis
<b>Calidad del software (bugs y reprocesos)</b>	Número de errores por nivel y cantidad de reprocesos.	Permite evaluar el impacto de las pruebas y la estabilidad del software.

<b>Tiempo de ciclo y velocidad de entrega</b>	Tiempo total de desarrollo y despliegue hasta producción.	Mide la eficiencia del flujo CI/CD y la capacidad de respuesta al negocio.
<b>Fallos en cambios y recuperación</b>	Frecuencia de fallos y tiempo para restaurar el servicio.	Refleja la confiabilidad operativa del proceso de entrega.
<b>Relación mejoras vs. bugs</b>	Comparación entre nuevas funcionalidades y errores reportados.	Indica si el sistema evoluciona con valor real o solo en correcciones.
<b>Nivel de automatización</b>	Porcentaje de pruebas, despliegues y validaciones automatizadas.	Reduce errores humanos y mejora la repetibilidad del proceso.
<b>Eficiencia del pipeline y pruebas</b>	Desempeño del pipeline CI/CD y efectividad de las pruebas.	Asegura entregas más estables y confiables.
<b>Tiempo de respuesta al usuario</b>	Tiempo de completación de procesos del usuario.	Mide el impacto directo del software en la operación del cliente.

Nota: Elaboración propia.

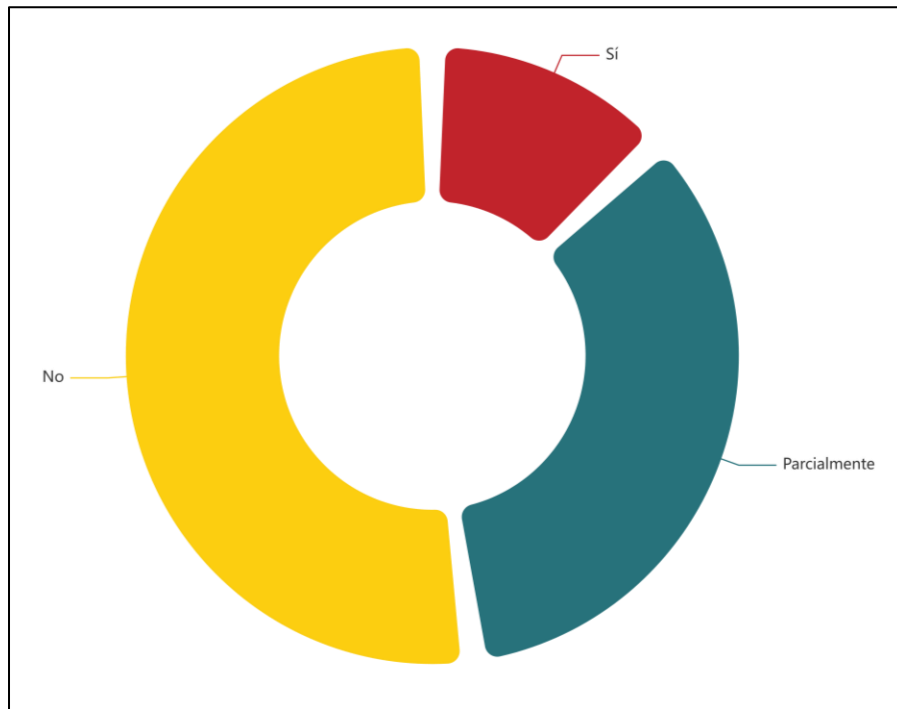
## **ENCUESTA - IMPLEMENTACIÓN DE INDICADORES Y MÉTRICAS:**

Para analizar la implementación de indicadores y métricas como complemento de los KPIs en la consultora NGS, se aplicó la encuesta correspondiente a esta sección, con el propósito de identificar el nivel de uso, seguimiento y aprovechamiento de métricas para la gestión del desempeño. Esta evaluación permite conocer en qué medida la organización cuenta con instrumentos formales para medir la eficiencia operativa, la calidad del software, la estabilidad de los despliegues y la mejora continua del servicio, así como establecer una base objetiva para fortalecer la toma de decisiones estratégicas alineadas con las prácticas DevOps y los objetivos institucionales.

A continuación, se presentan y describen las preguntas y respuestas desarrolladas para la sección de complemento de KPIs: Implementación de indicadores y métricas:

1. ¿Existen tableros o reportes que muestren indicadores de desempeño (DORA metrics, KPIs)?

**Figura 85** Disponibilidad de tableros o reportes con indicadores de desempeño (KPIs y métricas DORA).

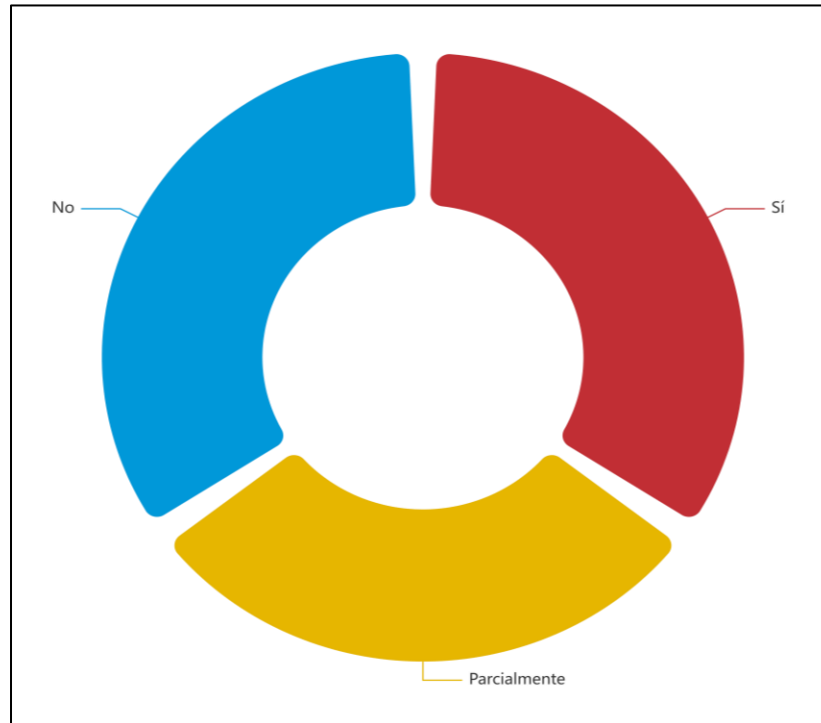


Nota: Elaboración propia.

El gráfico se muestra que no existe una implementación consolidada de tableros o reportes formales de indicadores de desempeño (KPIs o DORA metrics) en NGS, predominando la percepción de que estos solo se utilizan de manera parcial o no están disponibles. Esto refleja que la medición del desempeño aún no está institucionalizada ni estandarizada dentro de la organización, limitando la visibilidad del comportamiento real del flujo de entrega continua. La presencia de respuestas en la categoría “Parcialmente” sugiere esfuerzos aislados de seguimiento, pero sin un enfoque estructurado. En conclusión, esta situación representa una debilidad clave para la mejora continua, ya que dificulta la toma de decisiones basada en datos; por ello, se vuelve necesario implementar de forma formal tableros de control con KPIs y métricas DORA que permitan evaluar la eficiencia, calidad y estabilidad del proceso CI/CD en NGS.

2. ¿El equipo revisa periódicamente los indicadores de desempeño (por ejemplo, tiempo de entrega, tasa de fallos o frecuencia de despliegues)?

**Figura 86** *Revisión periódica de los indicadores de desempeño por parte del equipo.*

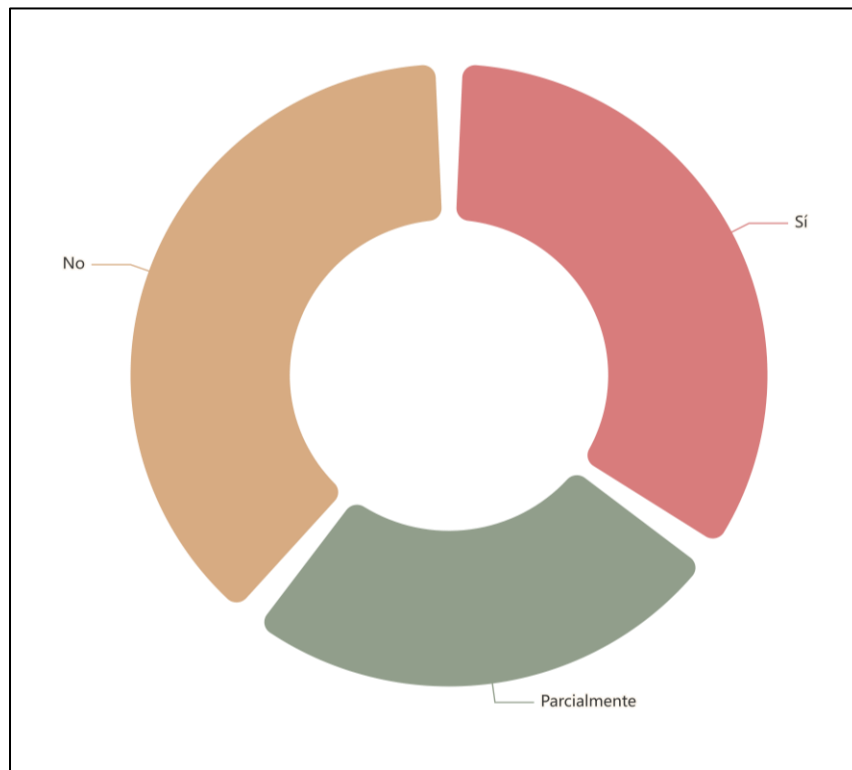


Nota: Elaboración propia.

Los resultados sobre la revisión periódica de los indicadores de desempeño (como tiempo de entrega, tasa de fallos o frecuencia de despliegues) muestran que las respuestas se distribuyen entre las opciones “Sí”, “Parcialmente” y “No”, lo que evidencia que esta práctica no se realiza de manera uniforme dentro de NGS. La presencia de un grupo que indica “Sí” refleja que en algunos equipos existe una cultura incipiente de seguimiento de indicadores; sin embargo, las respuestas “Parcialmente” y “No” confirman que dicho monitoreo no está formalizado ni estandarizado a nivel organizacional. En conjunto, los resultados indican que la revisión de KPIs se ejecuta de forma irregular y dependiente de iniciativas aisladas, más que como parte de un proceso institucionalizado. En conclusión, esta situación limita la capacidad de NGS de manera sistemática, su desempeño y fortalecer la mejora continua, por lo que se vuelve necesaria la formalización periódica de la revisión de indicadores como práctica clave dentro del modelo DevOps.

3. ¿Los resultados de los indicadores son utilizados para identificar oportunidades de mejora o tomar decisiones de gestión?

**Figura 87** *Uso de los indicadores para identificar mejoras o apoyar decisiones de gestión.*

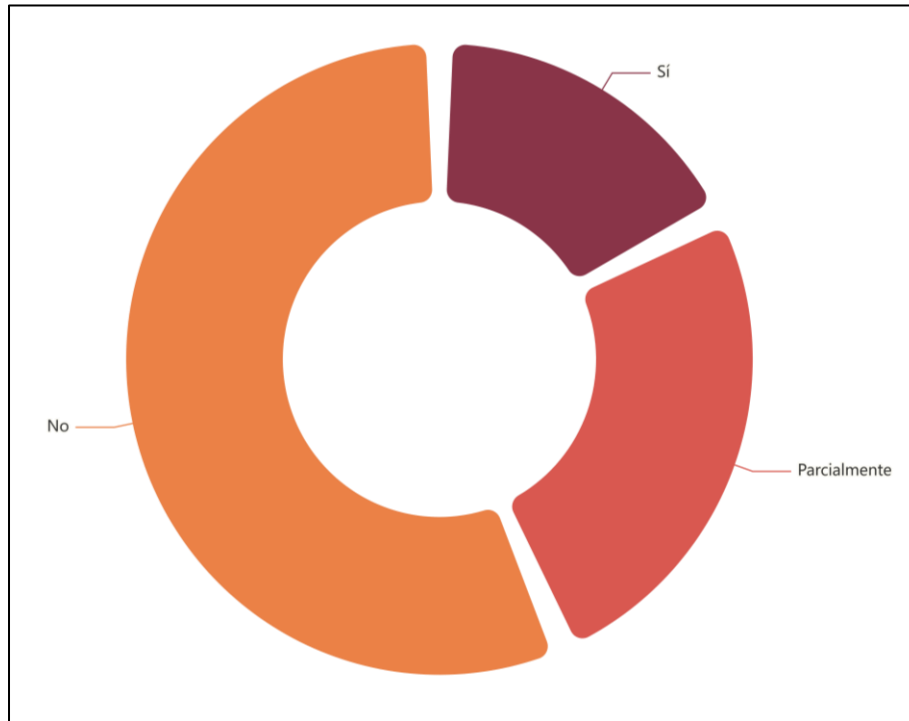


Nota: Elaboración propia.

El gráfico sobre el uso de los resultados de los indicadores para identificar oportunidades de mejora o apoyar la toma de decisiones de gestión muestra que las respuestas se distribuyen entre las opciones “Sí”, “Parcialmente” y “No”, lo que refleja que esta práctica no está completamente consolidada en NGS. La presencia de respuestas afirmativas indica que en ciertos casos los indicadores sí son considerados como insumo para la mejora y la gestión; sin embargo, el peso significativo de las opciones “Parcialmente” y “No” evidencia que el uso de métricas todavía es limitado, inconsistente y dependiente de iniciativas puntuales, más que de una política organizacional formal. En conjunto, los resultados permiten concluir que NGS aún no aprovecha de manera sistemática el potencial de los KPIs como herramienta estratégica de gestión, lo que restringe la identificación estructurada de oportunidades de mejora y debilita la toma de decisiones basada en datos, siendo necesario fortalecer esta práctica dentro del modelo de mejora continua.

4. ¿Las métricas DORA (frecuencia de despliegue, tiempo de recuperación, tasa de fallos y lead time) se miden actualmente en NGS?

**Figura 88** *Medición actual de las métricas DORA en NGS.*



Nota: Elaboración propia.

Los resultados sobre la medición de las métricas DORA en NGS (frecuencia de despliegue, tiempo de recuperación, tasa de fallos y lead time) muestran que predomina la respuesta “No”, seguida por “Parcialmente”, mientras que una proporción menor indica “Sí”. Este comportamiento evidencia que las métricas DORA no se miden de forma sistemática ni estandarizada en la organización, aunque existen intentos aislados de medición en algunos proyectos o equipos. La presencia de la categoría “Parcialmente” sugiere que ciertos indicadores se registran de manera informal o limitada, sin una integración completa a un sistema de gestión del desempeño. En conclusión, NGS se encuentra en una etapa inicial de madurez en la adopción de métricas DORA, lo que limita la capacidad de evaluar objetivamente la eficiencia del flujo CI/CD; por ello, resulta prioritario institucionalizar su medición para fortalecer la mejora continua y la toma de decisiones basada en datos.

#### **ENCUESTA: EVALUACIÓN DEL SERVICIO RECIBIDO POR CLIENTES EXTERNOS:**

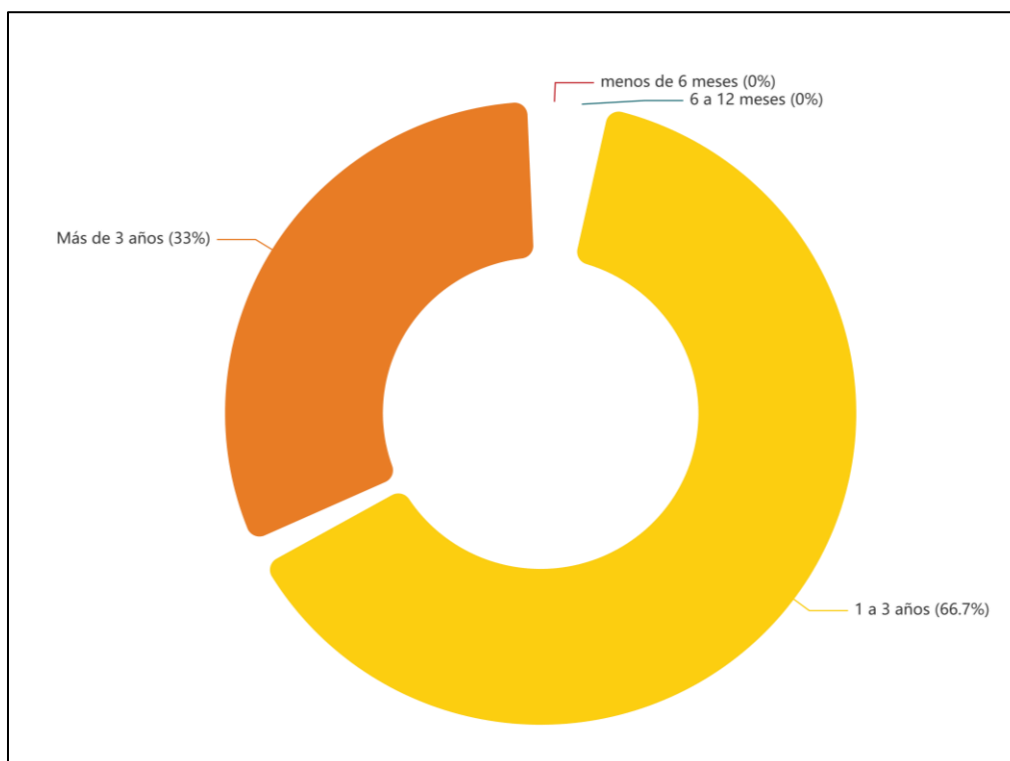
Se aplicó la encuesta, orientada a evaluar aspectos clave asociados a la entrega continua de software y la atención recibida. Esta sección permite valorar elementos como la claridad en la planificación y cumplimiento de plazos, la calidad del software entregado, la velocidad y eficiencia

en los despliegues, la comunicación durante el ciclo de entrega, la capacidad de respuesta ante incidencias, la transparencia de los procesos y métricas, así como el nivel de profesionalismo del equipo de NGS. Asimismo, incorpora preguntas abiertas que permiten identificar oportunidades de mejora y los aspectos más valorados por los clientes. La información recopilada constituye un insumo fundamental para evaluar el impacto real de los procesos internos desde la perspectiva del usuario final y orientar acciones de mejora continua del servicio.

A continuación, se presenta la sección de datos demográficos, donde se muestran las características de los participantes de la encuesta a través de gráficos con su respectivo detalle:

1. Tiempo de relación con NGS:

**Figura 89** *Tiempo de relación del cliente externo con NGS.*



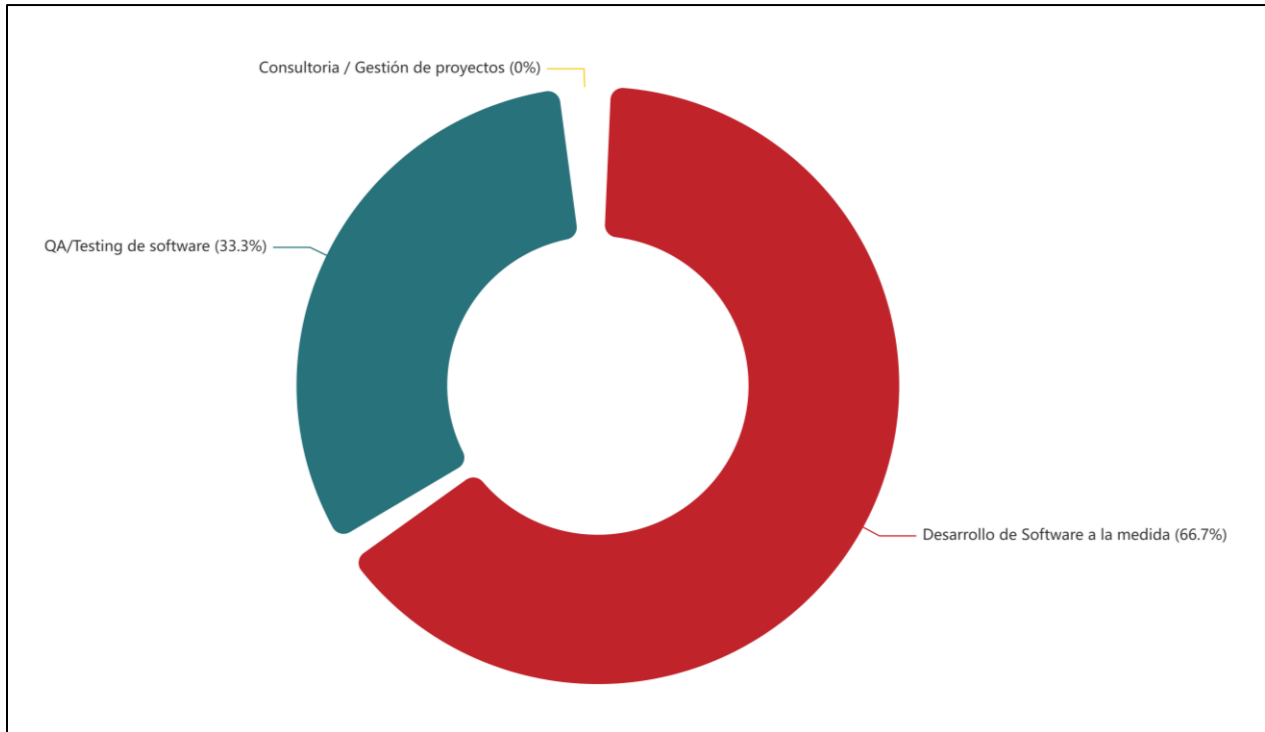
Nota: Elaboración propia.

La muestra de los participantes señala que el tiempo de relación de los clientes con NGS refleja que el 66.7 % (2 de 3 clientes) mantiene una relación de entre 1 a 3 años, mientras que el 33.3 % (1 de 3 clientes) posee una relación de más de 3 años. No se registran clientes con una relación menor a un año. Estos resultados evidencian que la mayoría de los clientes encuestados cuenta con una experiencia prolongada con NGS, lo que permite que su percepción del servicio

esté basada en un conocimiento continuo y sostenido de los procesos, la calidad del software y la entrega de soluciones. Esto también sugiere un nivel de fidelización positivo hacia la consultora.

## 2. Tipo de proyecto recibido:

**Figura 90** Tipo de proyecto recibido por el cliente externo.

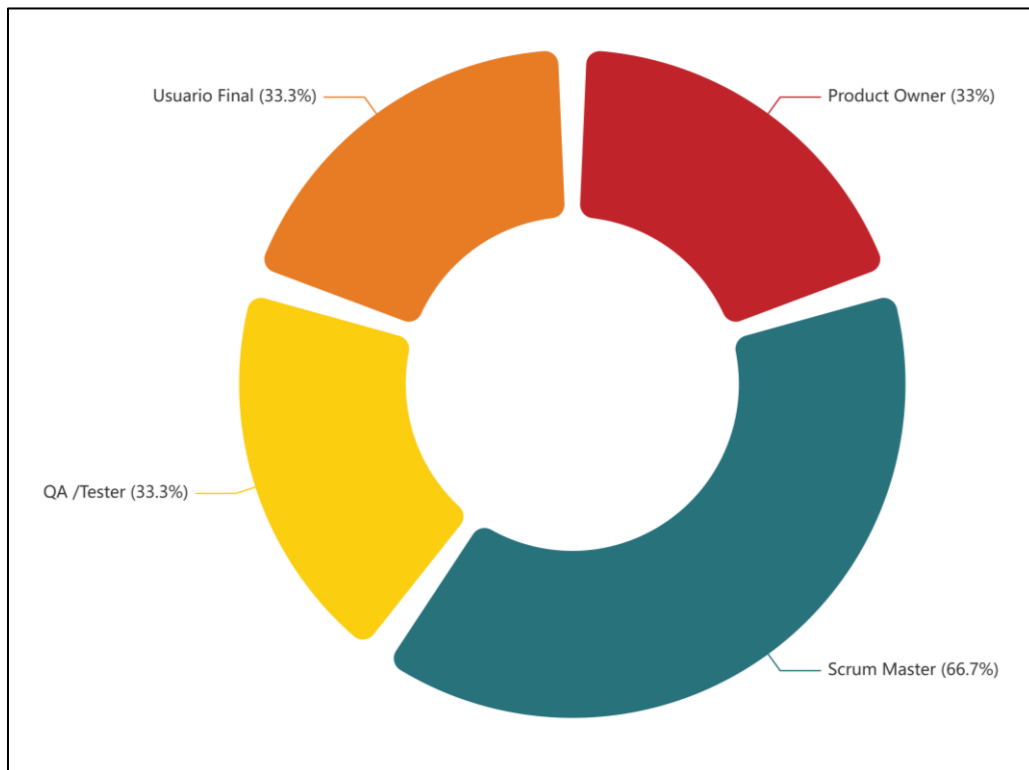


Nota: Elaboración propia.

El tipo de proyecto recibido por los clientes externos refleja que el 66.7 % (2 de 3 clientes) recibió servicios de desarrollo de software a la medida, mientras que el 33.3 % (1 de 3 clientes) recibió servicios de QA/Testing de software. No se registraron respuestas en la categoría de consultoría o gestión de proyectos. Estos resultados confirman que la encuesta fue aplicada a clientes con experiencia directa en los principales servicios ofrecidos por NGS, lo que aporta validez a la evaluación del servicio desde una perspectiva real y operativa.

## 3. Rol con el que participó en el proyecto:

**Figura 91** Rol desempeñado por el cliente en el proyecto.

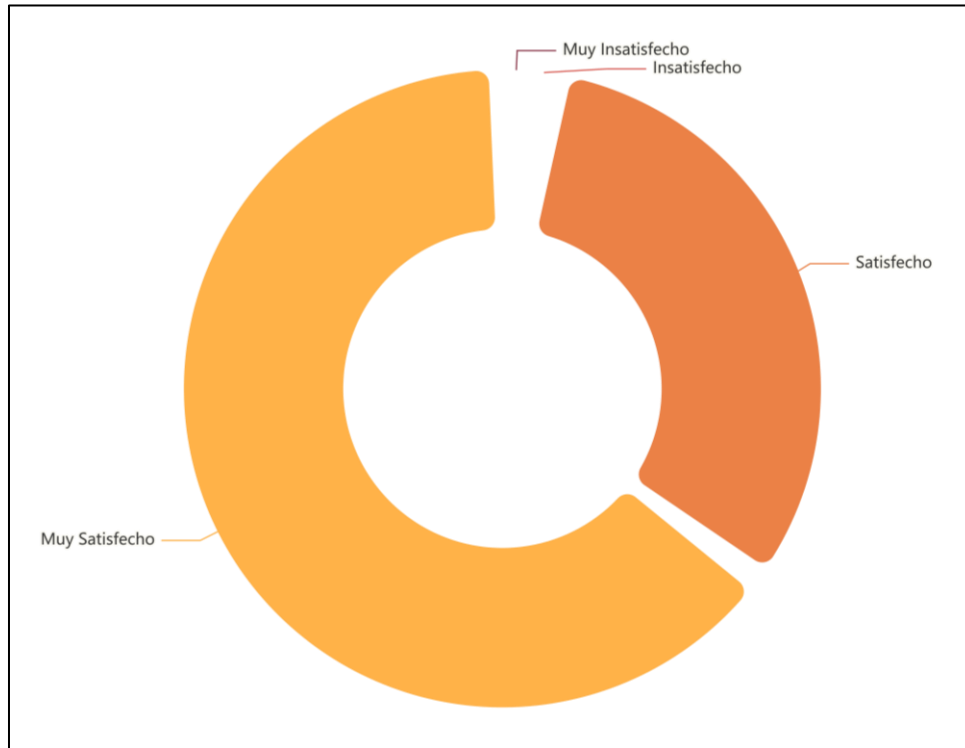


Nota: Elaboración propia.

Los roles de los participantes que se involucraron de los clientes en los proyectos de NGS muestran que el 66.7 % (2 de 3 clientes) desempeñó el rol de Scrum Master, lo que evidencia una participación en la gestión ágil y el seguimiento del proceso de desarrollo. Asimismo, el 33.3 % (1 de 3 clientes) participó como Product Owner, el 33.3 % (1 de 3 clientes) como QA/Tester y el 33.3 % (1 de 3 clientes) como usuario final. Estos resultados indican que la evaluación del servicio fue realizada desde múltiples perspectivas clave del ciclo de desarrollo, lo que fortalece la validez de la opinión recopilada al integrar enfoques estratégicos, técnicos y de uso final del producto.

4. Claridad en la planificación y cumplimiento de plazos de entrega:

**Figura 92** Claridad en la planificación y cumplimiento de plazos de entrega.

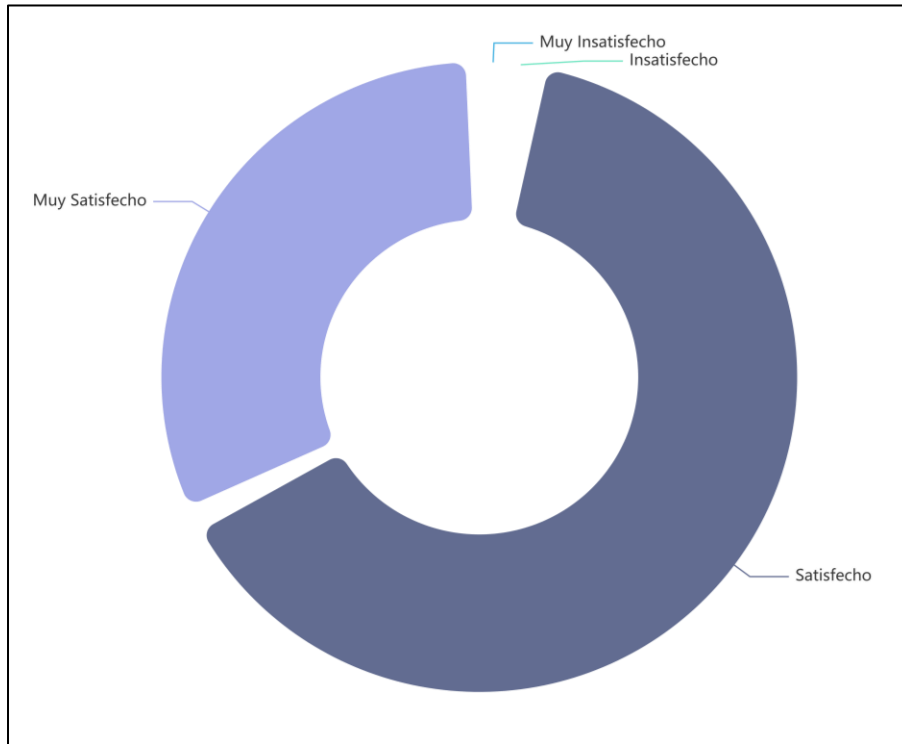


Nota: Elaboración propia.

Los resultados obtenidos evidencian una percepción mayoritariamente positiva por parte de los clientes externos respecto a la claridad en la planificación y el cumplimiento de los plazos de entrega por parte de NGS, ya que predominan las valoraciones de “Muy satisfecho” y “Satisfecho”. Esto indica que, en términos generales, los clientes consideran que la organización presenta un adecuado nivel de organización, comunicación y responsabilidad en la gestión de los tiempos de entrega. La presencia mínima de valoraciones negativas sugiere que los casos de insatisfacción son aislados y no representan una tendencia significativa. En conclusión, NGS demuestra un buen desempeño en la planificación y cumplimiento de plazos, lo que fortalece la confianza del cliente y respalda la efectividad de sus procesos de gestión y entrega continua, aunque siempre existen oportunidades de mejora para mantener y elevar este nivel de satisfacción.

5. Calidad del software entregado (funcionalidad, estabilidad, cumplimiento de requisitos):

**Figura 93** Calidad del software entregado (funcionalidad, estabilidad y requisitos).

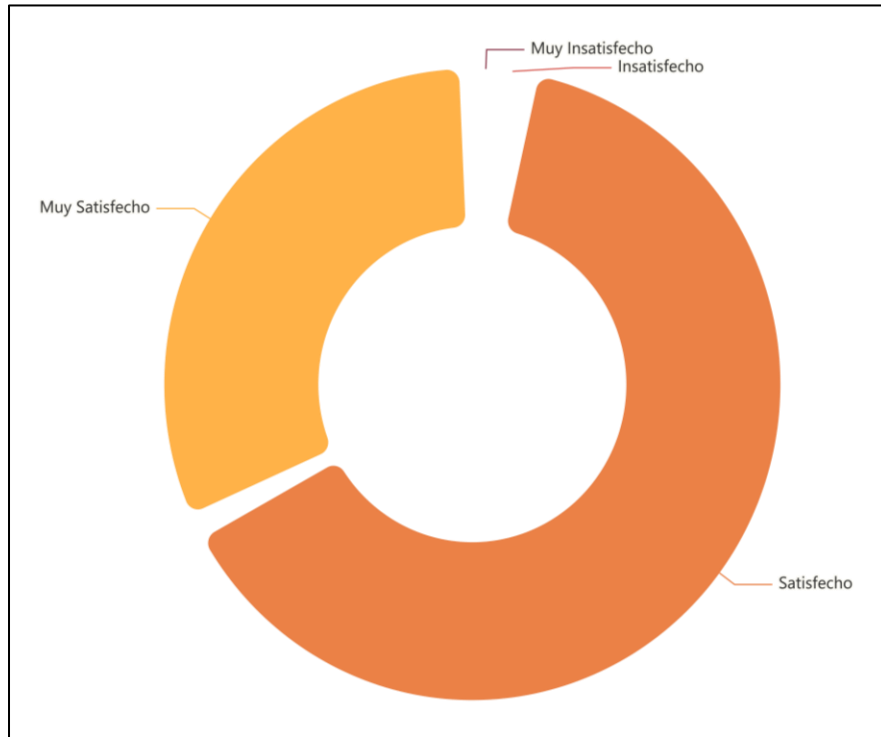


Nota: Elaboración propia.

El gráfico refleja que la percepción de los clientes externos sobre la calidad del software entregado por NGS es ampliamente positiva, ya que predominan las valoraciones de “Satisfecho” y “Muy satisfecho”. Esto indica que, en términos generales, los sistemas desarrollados cumplen con los requisitos funcionales establecidos, presentan un nivel adecuado de estabilidad y responden a las necesidades planteadas por los usuarios. La ausencia o mínima presencia de valoraciones negativas sugiere que los casos de insatisfacción son aislados. En conclusión, los resultados evidencian que NGS mantiene un buen nivel de calidad en sus productos de software, lo cual fortalece la confianza del cliente y respalda la efectividad de sus procesos de desarrollo y aseguramiento de la calidad dentro del enfoque de entrega continua.

6. Velocidad y eficiencia en los despliegues (optimización de la entrega continua):

**Figura 94** *Velocidad y eficiencia en los despliegues (optimización de la entrega continua).*

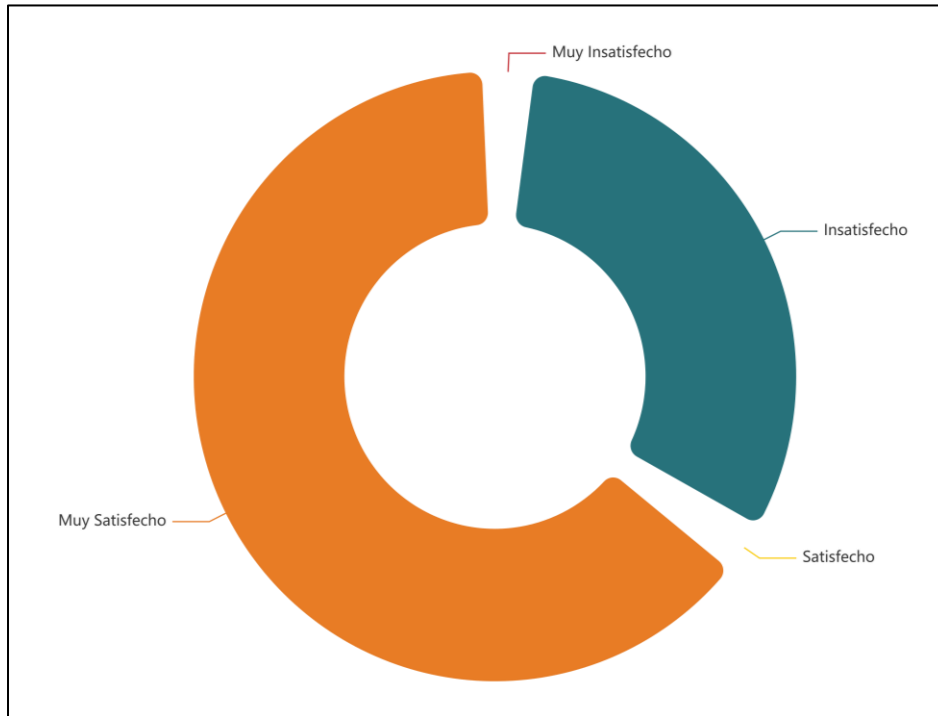


Nota: Elaboración propia.

El resultado de los clientes externos respecto a la velocidad y eficiencia de los despliegues en NGS es mayoritariamente positivo, predominando las valoraciones de “Satisfecho” y “Muy satisfecho”. Esto indica que, en términos generales, los clientes consideran que los procesos de entrega y liberación de software se realizan de forma oportuna y eficiente, contribuyendo a una adecuada optimización de la entrega continua. La presencia mínima de valoraciones negativas sugiere que los retrasos o dificultades en los despliegues son casos aislados y no representan una tendencia general. En conclusión, NGS presenta un buen desempeño en la eficiencia de sus despliegues, lo que respalda la efectividad de sus prácticas actuales de entrega continua; no obstante, siempre existe la oportunidad de fortalecer aún más la automatización para incrementar la rapidez, estabilidad y confiabilidad del proceso.

#### 7. Comunicación y seguimiento durante el ciclo de entrega:

**Figura 95** Comunicación y seguimiento durante el ciclo de entrega.

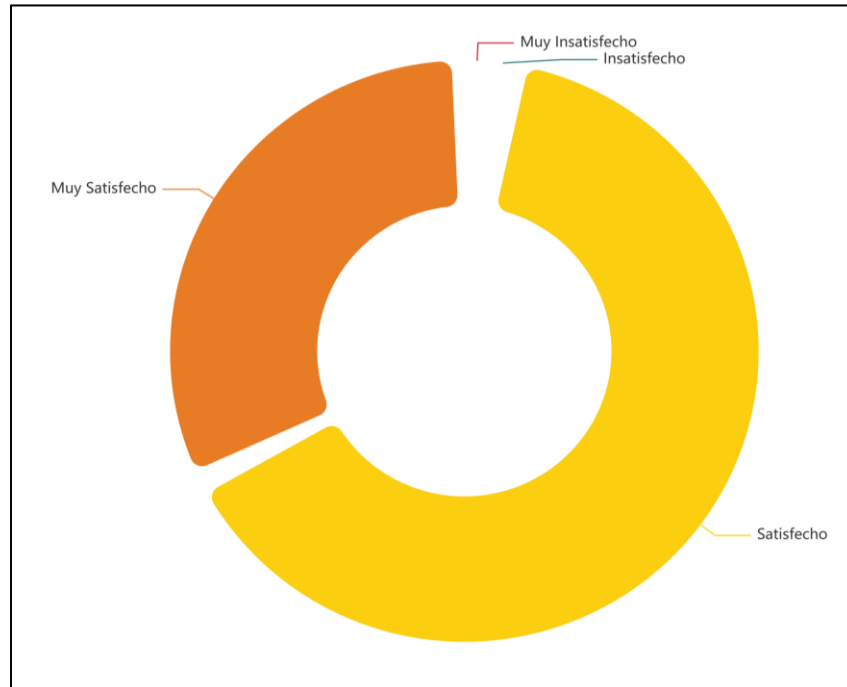


Nota: Elaboración propia.

Las respuestas evidencian una tendencia general de satisfacción en relación con la comunicación y el seguimiento durante el ciclo de entrega en NGS, lo que indica que los clientes perciben un acompañamiento adecuado a lo largo del desarrollo de los proyectos. La claridad en la información, el seguimiento de avances y la disponibilidad del equipo son aspectos que influyen positivamente en la experiencia del cliente. Aunque existen algunas percepciones de insatisfacción, estas son minoritarias y no representan un patrón generalizado de deficiencias. En conjunto, la comunicación se consolida como una fortaleza operativa de NGS, ya que contribuye a fortalecer la confianza, la transparencia del proceso y la relación con los clientes. No obstante, se identifican oportunidades para estandarizar aún más los mecanismos de seguimiento, de manera que todos los proyectos mantengan un nivel de comunicación homogéneo y alineado con las mejores prácticas de la gestión ágil y la entrega continua.

8. Capacidad de respuesta ante incidencias o cambios solicitados:

**Figura 96** Capacidad de respuesta ante incidencias o cambios solicitados.

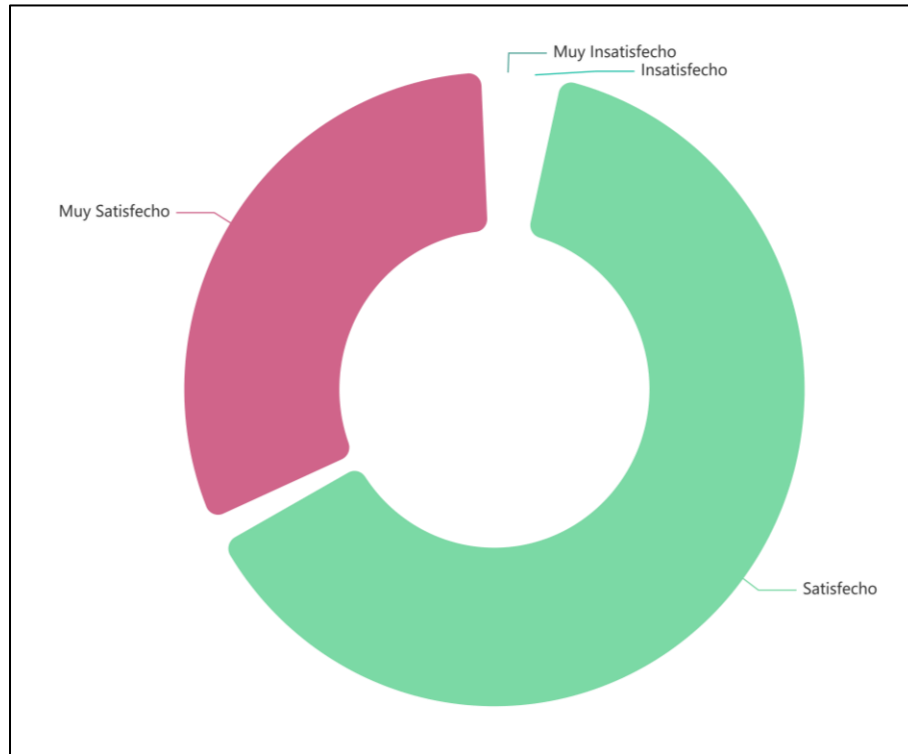


Nota: Elaboración propia.

Las respuestas reflejan una percepción mayoritariamente positiva en cuanto a la capacidad de respuesta de NGS ante incidencias o cambios solicitados, predominando los niveles de satisfacción. Los clientes reconocen que el equipo muestra disposición, atención oportuna y seguimiento adecuado cuando se presentan ajustes o situaciones imprevistas durante el desarrollo de los proyectos. La presencia reducida de valoraciones de insatisfacción indica que los casos negativos son aislados y no representan una tendencia generalizada. En conjunto, este resultado evidencia que la gestión de incidencias y cambios constituye una fortaleza operativa para NGS, ya que impacta favorablemente en la confianza del cliente y en la continuidad del servicio. No obstante, se identifican oportunidades de mejora orientadas a estandarizar aún más los tiempos de respuesta y los mecanismos de atención, con el fin de garantizar un nivel de servicio homogéneo en todos los proyectos.

9. Transparencia en los procesos y métricas compartidas (KPIs, avances, informes):

**Figura 97** *Transparencia en los procesos y métricas compartidas con el cliente.*

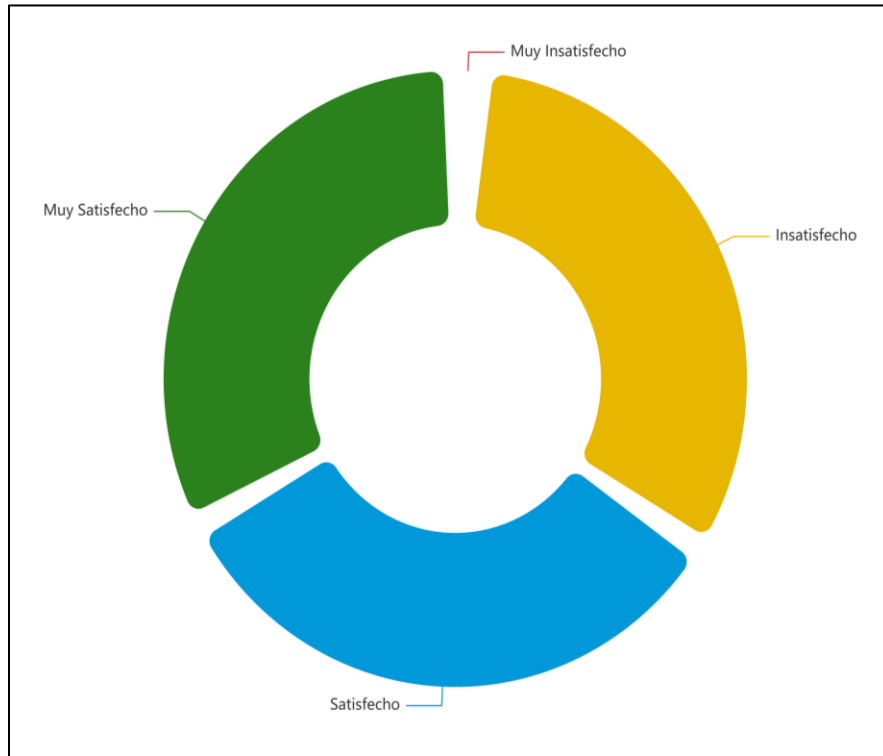


Nota: Elaboración propia.

Las respuestas reflejan una percepción mayoritariamente favorable respecto a la transparencia en los procesos y en la información compartida por NGS, especialmente en lo relacionado con avances de proyectos e informes de seguimiento. Los niveles de satisfacción predominantes indican que los clientes consideran que existe un grado adecuado de apertura y comunicación sobre el estado de los trabajos. Las valoraciones de insatisfacción son mínimas, lo que sugiere que las debilidades en este aspecto son casos puntuales y no una práctica generalizada. En conjunto, la transparencia se consolida como una fortaleza en la relación con los clientes, ya que fortalece la confianza y la credibilidad del servicio. No obstante, se identifican oportunidades para fortalecer la formalización y estandarización en el uso y divulgación de KPIs, de modo que la información sea aún más clara, sistemática y útil para la toma de decisiones.

10. Atención recibida por parte del equipo de NGS (profesionalismo y disposición):

**Figura 98** Atención recibida por parte del equipo de NGS (profesionalismo y disposición).

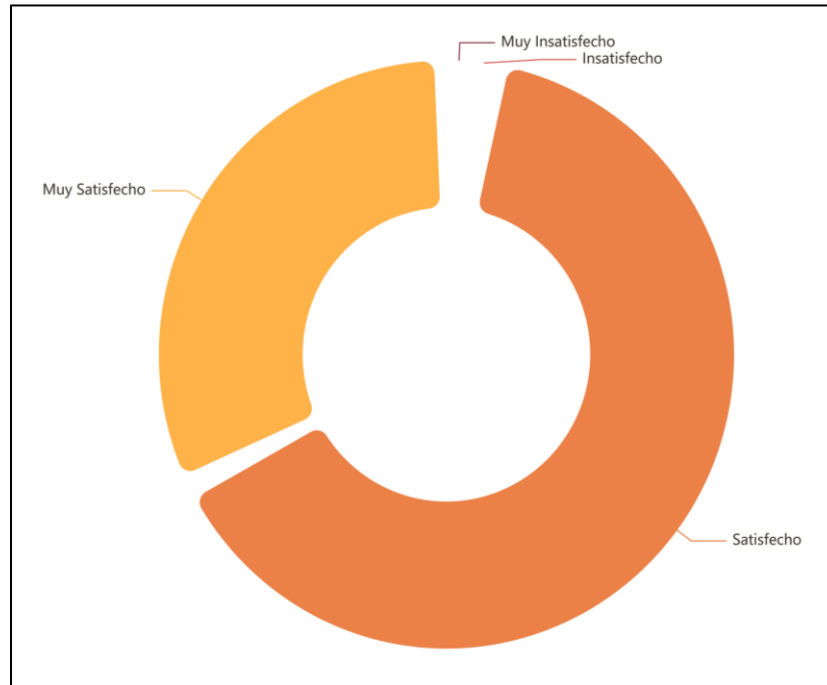


Nota: Elaboración propia.

Los resultados muestran una valoración general positiva sobre la atención brindada por el equipo de NGS, destacándose percepciones de satisfacción asociadas al profesionalismo, la disposición y el trato recibido durante el desarrollo de los proyectos. Los clientes reconocen la actitud colaborativa del equipo, la apertura para atender requerimientos y la responsabilidad en el acompañamiento del servicio. Aunque se identifican algunas percepciones de insatisfacción, estas son minoría y no representan un patrón generalizado. En conjunto, la atención al cliente se consolida como una fortaleza institucional de NGS, ya que impacta directamente en la confianza, la calidad de la relación y la imagen profesional de la organización; no obstante, se visualiza la oportunidad de seguir fortaleciendo estas competencias de manera homogénea en todos los equipos.

#### 11. Nivel de satisfacción general con los servicios de NGS:

**Figura 99** Nivel de satisfacción general con los servicios de NGS



Nota: Elaboración propia.

El resultado refleja un alto nivel de satisfacción general por parte de los clientes con respecto a los servicios brindados por NGS, ya que predominan claramente las valoraciones de “Satisfecho” y “Muy satisfecho”. Esto indica que, en términos globales, los clientes perciben que la organización cumple con sus expectativas en aspectos como calidad del software, cumplimiento de plazos, atención recibida y comunicación durante los proyectos. Las valoraciones de insatisfacción son mínimas, lo que sugiere que los casos de inconformidad son aislados y no representan una tendencia significativa. En conjunto, este resultado evidencia que NGS goza de una imagen favorable ante sus clientes, lo que fortalece su posicionamiento como proveedor de servicios de software confiable; no obstante, persiste la oportunidad de seguir perfeccionando sus procesos para sostener y elevar este nivel de satisfacción en el tiempo.

12. Responda abiertamente: Desde su experiencia, ¿Qué aspectos considera que se deben mejorar en la entrega continua de software?

**Tabla 41** Aspectos que deben mejorarse según la experiencia del cliente en la entrega continua de software.

Aspecto clave	Descripción breve	Análisis
<b>Madurez en la entrega continua</b>	Se solicita mayor madurez y mejor puesta en práctica del proceso.	Indica la necesidad de fortalecer la automatización, estandarización y estabilidad del flujo CI/CD para lograr mayor confiabilidad en las entregas.
<b>Diseño gráfico del software</b>	Se sugiere mejorar el diseño visual de las aplicaciones.	Refleja oportunidades de mejora en la experiencia de usuario (UX/UI), lo que impacta directamente en la percepción de calidad del producto.

Nota: Elaboración propia.

13. Responda abiertamente: Desde su experiencia, ¿Qué aspecto valora más de la experiencia con NGS?

El principal aspecto valorado por los clientes en su experiencia con NGS es el compromiso del equipo de trabajo, el cual se refleja en la responsabilidad, disposición y cumplimiento constante a lo largo del servicio prestado. Esta actitud genera confianza, fortalece la relación con el cliente y contribuye de manera significativa a mantener de forma positiva la calidad del servicio, y mostrando que el factor humano constituye uno de los pilares más sólidos en la experiencia con NGS.

Los resultados evidencian que en NGS existen avances operativos percibidos, especialmente en tiempos de entrega, estabilidad del software y frecuencia de despliegues. Sin embargo, dichos avances no están respaldados por un sistema formal de indicadores institucionalizado. El personal técnico reconoce la importancia de los KPIs, pero presenta un conocimiento parcial de métricas clave como las métricas DORA, lo que limita su aplicación estratégica. También, se identificó que la disponibilidad de tableros de control es limitada, no existe una revisión periódica formal de indicadores, y el uso de métricas para la toma de decisiones gerenciales es aún incipiente. Esto refleja que la entrega continua en NGS se basa más en la experiencia del equipo que en una gestión estructurada por datos.

Desde una perspectiva de los clientes externos, los resultados muestran una satisfacción general positiva en cuanto a calidad del software, atención recibida, comunicación y capacidad de respuesta ante incidencias. No obstante, los clientes manifiestan oportunidades de mejora en cumplimiento de plazos, optimización de tiempos de entrega y mayor transparencia en el uso de

métricas de desempeño.

**Tabla 42** Hallazgos sobre la ausencia de KPIs y su impacto en la entrega de valor CI/CD en NGS

No	Dimensión	Hallazgo	Consecuencia en la Entrega
1	Sistema de KPIs	No existe un sistema formal e institucionalizado de KPIs y métricas DORA.	Los entregables no se controlan de forma objetiva ni medible.
2	Conocimiento de métricas	El personal presenta desconocimiento parcial de los indicadores clave.	Se dificulta la correcta interpretación del desempeño del pipeline CI/CD.
3	Tableros de control	No existen tableros de indicadores centralizados y automatizados.	Baja visibilidad del estado real de la entrega.
4	Revisión de indicadores	No se realiza seguimiento periódico y sistemático de los KPIs.	Los entregables se gestionan de forma reactiva y no preventiva.
5	Tiempos de entrega	Las mejoras en tiempos no están respaldadas por datos consolidados.	No se puede validar objetivamente la eficiencia del CI/CD.
6	Calidad del software	La calidad se evalúa principalmente por percepción y no por métricas técnicas.	Riesgo de fallas no detectadas oportunamente.
7	Frecuencia de despliegues	No se controlan formalmente los despliegues mediante indicadores.	Falta de control sobre la estabilidad del entorno productivo.
8	Uso de KPIs en decisiones	Las decisiones se toman sin respaldo sistemático en datos.	Riesgo de decisiones erróneas que afecten los entregables.

9	Transparencia con el cliente	No se comparten métricas de desempeño de forma clara con el cliente.	Debilita la confianza y el control del servicio entregado.
10	Cultura de medición	No existe una cultura consolidada de medición del desempeño.	La entrega de las solicitudes depende más del esfuerzo humano que de un modelo gestionado.

Nota: Elaboración propia.

Luego de analizar la situación actual de NGS respecto al uso de indicadores de desempeño, se evidenció la ausencia de un sistema formal de medición, desconocimiento parcial del personal sobre métricas clave y falta de tableros automatizados. Estos hallazgos revelan una necesidad urgente de establecer lineamientos técnicos que permitan evaluar objetivamente el desempeño del flujo CI/CD en la organización.

Como complemento al diagnóstico de la situación actual de NGS respecto al uso de indicadores de desempeño, se realizó una revisión documental centrada en la identificación de métricas ampliamente utilizadas en la industria para evaluar la madurez y el impacto de la adopción de DevOps. Esta revisión tuvo como objetivo contrastar los hallazgos obtenidos con estándares reconocidos y validados empíricamente.

Entre las métricas identificadas, destacan las métricas DORA (DevOps Research and Assessment), validadas en investigaciones recientes como la tesis de Aguilar Riera y Naula Sanisaca (2025), que confirman su utilidad para medir el rendimiento de equipos de desarrollo a través de cuatro métricas fundamentales:

A continuación se presentan las métricas identificadas en la revisión:

**Figura 100** Métricas DORA



Nota: Elaboración propia.

La siguiente tabla muestra cómo las métricas DORA se relacionan con los tiempos de entrega, la calidad del software y la satisfacción del cliente, destacando oportunidades clave de mejora para NGS según el diagnóstico realizado.

**Tabla 43** Métricas DORA y su relación con las dimensiones del Objetivo 4

Métrica DORA	Descripción	Dimensión	Potencial de mejora en NGS
Deployment Frequency	Frecuencia de despliegues a producción.	Tiempos de entrega	Incrementar automatización para liberar cambios más seguido.
Lead Time for Changes	Tiempo desde commit hasta despliegue.	Tiempos de entrega	Optimizar el flujo CI/CD reduciendo tiempos de espera y prueba.
Change Failure Rate	Porcentaje de cambios que causan errores en producción.	Calidad del software	Mejorar pruebas automatizadas y control de calidad.
Mean Time to Recovery (MTTR)	Tiempo medio para restaurar el sistema tras una falla.	Calidad del software / Satisfacción del cliente	Fortalecer monitoreo y respuesta ante incidentes.

Nota: Elaboración propia basado de Aplicación de principios DEVOPS y metodologías ágiles para el desarrollo del módulo del historial financiero de la Cooperativa de Ahorro y Crédito Jardín Azuayo.

La adopción de las métricas DORA resulta altamente pertinente para NGS, ya que ofrecen un marco estandarizado y validado internacionalmente para medir el desempeño del flujo DevOps en dimensiones críticas como velocidad, calidad y resiliencia. Estas métricas frecuencia de despliegue, tiempo de entrega de cambios, tasa de fallos en producción y tiempo medio de recuperación permiten transformar percepciones subjetivas en indicadores objetivos, facilitando la toma de decisiones basadas en datos. Su incorporación permitiría a NGS evaluar de forma continua la eficiencia de sus procesos, detectar cuellos de botella, y fortalecer la transparencia con los clientes mediante reportes claros y medibles. Esta evaluación preliminar sienta las bases para una gestión del desempeño más madura, alineada con las mejores prácticas de la industria.

Luego de este análisis de las métricas DORA, se presentan los principales hallazgos para este objetivo:

**Tabla 44** *Hallazgos sobre la gestión de KPIs y el potencial de adopción de métricas DORA en NGS*

No	Hallazgo	Impacto desde la perspectiva DevOps
1	Ausencia de un sistema formal de KPIs	Impide evaluar de forma objetiva la eficiencia del flujo DevOps y limita la mejora continua.
2	Desconocimiento de métricas DORA	El personal no identifica indicadores clave como MTTR o frecuencia de despliegue, afectando la gestión del rendimiento.
3	Falta de herramientas de visualización	Sin dashboards o tableros automatizados, es difícil monitorear el pipeline en tiempo real.
4	Evaluación subjetiva de la calidad del software	La falta de métricas objetivas (ej. tasa de fallos) genera una visión poco precisa del producto entregado.
5	Implementar métricas DORA permitiría mejorar visibilidad y toma de decisiones	Su adopción ayudaría a establecer líneas base, detectar cuellos de botella y alinear la entrega continua con objetivos de negocio.

Nota: Elaboración propia.

Luego de la aplicación de los instrumentos de recolección de datos y del análisis correspondiente para cada uno de los cuatro objetivos específicos, se muestra una tabla resumen con los principales hallazgos identificados. Esta síntesis permite visualizar de forma estructurada

las brechas detectadas en NGS en relación con la adopción de prácticas DevOps, sirviendo como base para el diseño de la propuesta en los siguientes capítulos.

**Tabla 45** *Resumen de hallazgos por objetivo específico*

<b>Objetivo</b>	<b>Enfoque</b>	<b>Hallazgos principales</b>
1. Diagnosticar la situación actual de la empresa en relación con prácticas DevOps.	Evaluación cultural, organizacional y técnica.	<ul style="list-style-type: none"> <li>• Bajo conocimiento integral de DevOps.</li> <li>• Procesos de desarrollo y operaciones poco integrados.</li> <li>• Resistencia al cambio y baja colaboración interáreas.</li> </ul>
2. Analizar el nivel de madurez de la organización en prácticas DevOps y metodologías ágiles.	Evaluación de prácticas ágiles y automatización.	<ul style="list-style-type: none"> <li>• Niveles bajos a medios de madurez en automatización, entrega continua y colaboración.</li> <li>• Uso parcial de metodologías ágiles sin integración al flujo completo.</li> <li>• Débil adopción de herramientas colaborativas.</li> </ul>
3. Evaluar la arquitectura tecnológica para soportar CI/CD.	Escalabilidad, seguridad y trazabilidad.	<ul style="list-style-type: none"> <li>• Arquitectura monolítica con poca elasticidad.</li> <li>• Falta de seguridad integrada en el pipeline.</li> <li>• Trazabilidad fragmentada y débil visibilidad de extremo a extremo.</li> </ul>
4. Proponer indicadores clave de desempeño (KPIs) para evaluar el impacto de DevOps.	Tiempos de entrega, calidad y satisfacción.	<ul style="list-style-type: none"> <li>• Ausencia de sistema formal de KPIs.</li> <li>• Desconocimiento de métricas DORA.</li> <li>• Evaluación subjetiva de la calidad.</li> <li>• Falta de visualización e indicadores compartidos con clientes.</li> <li>• Potencial de mejora con implementación de métricas DORA.</li> </ul>

Nota: Elaboración propia.

## CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

En base a los resultados obtenidos a lo largo de la investigación, a partir del análisis de las brechas en procesos, herramientas y competencias, la identificación de prácticas y herramientas DevOps aplicables a NGS, así como la evaluación de la arquitectura tecnológica y los indicadores de desempeño, se establecen los elementos clave que explican la situación actual de la organización. Estas conclusiones integran la evidencia recopilada mediante encuestas, entrevistas, Checklist técnico y análisis documental, proporcionando una visión consolidada que orienta tanto la comprensión del escenario actual como las acciones estratégicas necesarias para avanzar hacia un modelo de entrega continua más eficiente, automatizado y alineado con las buenas prácticas internacionales.

1. Se concluye que NGS presenta brechas estructurales que limitan la adopción efectiva de un flujo CI/CD institucionalizado. Persisten actividades manuales en etapas clave como integración, pruebas y despliegue, lo cual incrementa el riesgo de errores y retrabajos. La estandarización de procesos es limitada y varía entre equipos. Las herramientas actuales no conforman un ecosistema DevOps integrado, y el conocimiento técnico presenta vacíos en automatización y gestión del pipeline. Estas debilidades afectan negativamente la trazabilidad, calidad y eficiencia de la entrega continua. NGS se encuentra en una fase de transición tecnológica que requiere acciones estratégicas para evolucionar hacia un modelo DevOps maduro.
2. Las prácticas y herramientas DevOps identificadas a través de los instrumentos desarrollados son aplicables al contexto de NGS y son compatibles con marcos de trabajo como ITIL V4, especialmente en procesos como gestión de cambios, despliegues y seguridad. No obstante, su implementación en NGS es parcial y no está estandarizada. Cada equipo adopta prácticas bajo su propio criterio, sin una gobernanza clara. Existen herramientas útiles como control de versiones y automatización básica, pero su uso no es uniforme ni está alineado con políticas organizacionales. Esta fragmentación impide maximizar los beneficios de DevOps y limita su impacto a nivel organizativo.
3. La arquitectura actual de NGS tiene una base funcional con buenas prácticas en diseño y uso de frameworks, pero no está completamente alineada con los requerimientos de un

flujo CI/CD moderno. Existen debilidades en automatización de despliegues, monitoreo, trazabilidad y uso de contenedores. El monitoreo y la disponibilidad no son uniformes, y no se cuenta con un repositorio institucional que centralice los activos arquitectónicos. Esto limita la visibilidad técnica, la gobernanza y la capacidad de escalar servicios de forma ágil y segura. Aunque existe compatibilidad con prácticas DevOps, la arquitectura necesita mejoras para garantizar confiabilidad, escalabilidad y sostenibilidad.

4. No se ha implementado un sistema estructurado de métricas que permita evaluar el rendimiento del pipeline CI/CD. A pesar de percepciones positivas sobre mejoras en tiempos de entrega, calidad del software y satisfacción del cliente, estas no se sustentan con datos objetivos. El desconocimiento parcial de métricas clave, como las DORA, y la falta de tableros automatizados limitan la gestión basada en evidencia. La toma de decisiones se fundamenta en intuición más que en indicadores. Esta carencia restringe la mejora continua, debilita la capacidad de anticipar fallos y afecta la confianza del cliente al no haber transparencia cuantitativa sobre el servicio entregado.

## **5.2 RECOMENDACIONES**

1. Diseñar e implementar un plan de cierre de brechas que contemple mejoras en procesos, herramientas y competencias. Este plan debe incluir la estandarización del flujo CI/CD, la integración de herramientas de automatización, pruebas y control de versiones. Se recomienda promover la formación técnica del personal en áreas clave como DevOps, infraestructura como código, pruebas continuas y contenedores. Se debe institucionalizar la documentación de procesos para reducir la dependencia del conocimiento individual y asignar responsables por cada fase del pipeline. El acompañamiento de la alta dirección será clave para asegurar sostenibilidad y monitoreo del progreso.
2. Formalizar un marco de políticas DevOps que regule el uso de herramientas, responsabilidades y procedimientos. Estas políticas deben alinearse con las prácticas de ITIL V4 para asegurar gobernanza y calidad del servicio. Se recomienda definir lineamientos sobre gestión de cambios, control de versiones, pruebas automatizadas y seguridad. Además, se sugiere institucionalizar la figura de responsables de gobernanza DevOps para velar por la implementación transversal del modelo. El enfoque debe ser progresivo, partiendo de pilotos hasta llegar a una adopción organizacional, respaldado por

capacitación técnica y métricas operativas.

3. Fortalecer la arquitectura técnica mediante la implementación de pipelines CI/CD estandarizados, monitoreo centralizado y despliegues automatizados. Se recomienda avanzar en el uso de contenedores, servicios en la nube y respaldo automático. La observabilidad y trazabilidad deben integrarse como pilares en el diseño arquitectónico. Se requiere establecer un repositorio arquitectónico institucional y mantener actualizada la documentación técnica. La alta disponibilidad y recuperación ante fallos deben aplicarse de manera progresiva a los sistemas críticos. Estas acciones mejorarán la estabilidad operativa y facilitarán una entrega confiable, escalable y segura.
4. Establecer un sistema institucionalizado de indicadores clave de desempeño orientados a la entrega continua. Se recomienda adoptar métricas DORA como frecuencia de despliegue, tiempo de entrega, tasa de fallos y tiempo medio de recuperación, ya que permiten evaluar la eficiencia y calidad del pipeline. Es fundamental desarrollar tableros de control automatizados y fomentar su revisión periódica. Además, capacitar al personal en la interpretación de KPIs y su aplicación en la toma de decisiones fortalecerá la cultura de mejora continua. La visibilidad y transparencia de estos indicadores hacia los clientes aumentará la confianza y facilitará el cumplimiento de acuerdos de servicio.
5. Se recomienda a la consultora Nueva Generación de Software (NGS) la creación formal de un Área DevOps, complementada con la incorporación de roles Site Reliability Engineering (SRE), como una estrategia para fortalecer la estandarización del flujo de integración y entrega continua (CI/CD), mejorar la confiabilidad de los servicios tecnológicos y reducir los tiempos de respuesta ante incidencias. Esta área deberá funcionar como una unidad transversal habilitadora, alineada con las metodologías ágiles y las prácticas de ITIL v4, permitiendo integrar automatización, control de versiones, gestión de incidentes y uso sistemático de métricas de desempeño (DORA). Se sugiere que su implementación cuente previamente con la aprobación del CEO y la alta dirección, e inicie con un proceso de capacitación para los equipos involucrados, garantizando así un cambio cultural progresivo, una adopción controlada del modelo propuesto y la sostenibilidad de la mejora continua en NGS.

## **CAPÍTULO VI - APLICABILIDAD**

### **6.1 MODELO DE INTEGRACIÓN DE PRÁCTICAS DEVOPS CON ENFOQUE ITIL V4 Y ALINEACIÓN ÁGIL MEDIANTE SCRUM PARA FORTALECER EL FLUJO CI/CD EN NGS**

#### **6.1.1 INTRODUCCIÓN**

Según los resultados obtenidos durante la investigación y el análisis de los instrumentos aplicados, se identificaron diversas necesidades dentro de NGS relacionadas con la estandarización de procesos, la automatización del flujo de desarrollo y despliegue, la mejora de competencias técnicas y la alineación de prácticas operativas con marcos internacionales como ITIL V4. Asimismo, los desafíos asociados a la falta de trazabilidad, el uso inconsistente de herramientas y la gestión reactiva del monitoreo evidencian áreas críticas que requieren fortalecimiento para avanzar hacia un modelo de entrega continua más eficiente, confiable y repetible.

Estos hallazgos demuestran que la implementación de un modelo integrado DevOps-CI/CD basado en ITIL V4 posee un alto potencial para mejorar la coordinación entre equipos, optimizar tiempos de entrega, reducir actividades manuales, fortalecer la calidad del software y promover una cultura organizacional orientada a la mejora continua. A partir de las capacidades actuales de NGS y de las brechas identificadas, esta propuesta plantea un conjunto de estrategias, procesos y herramientas que permiten conducir la transición hacia un entorno tecnológico más automatizado, controlado y alineado con las mejores prácticas internacionales.

El presente capítulo expone cómo esta propuesta puede ser implementada de manera efectiva, considerando los recursos existentes, las tecnologías disponibles y las estrategias de mitigación derivadas de la matriz FODA, con el fin de garantizar una adopción progresiva, sostenible y alineada a las necesidades reales de la organización.

#### **6.2 JUSTIFICACIÓN DE LA PROPUESTA**

La propuesta surge como respuesta a los hallazgos del diagnóstico institucional, donde se identificaron brechas críticas en procesos manuales, baja integración de herramientas, ausencia de indicadores clave y debilidades en la arquitectura tecnológica para un flujo CI/CD maduro. A ello se suma la falta de institucionalización de prácticas DevOps y la necesidad de reforzar la coordinación entre áreas técnicas.

Desde el enfoque teórico, la propuesta se sustenta en tres pilares: DevOps como práctica

de automatización y colaboración para la entrega continua; ITIL V4 como marco de gestión de servicios alineado a valor y gobernanza; y Scrum como marco ágil que promueve iteraciones cortas, adaptabilidad y trabajo en equipo.

Al integrar estos enfoques, se busca no solo mejorar la eficiencia técnica del flujo CI/CD, sino también institucionalizar prácticas de planificación ágil, revisión continua y transparencia del trabajo, alineando la propuesta con estándares internacionales y con las capacidades existentes de la consultora. Esta combinación metodológica garantiza un modelo sostenible, adaptable y medible.

## **6.3 ALCANCE DE LA PROPUESTA**

### **6.3.1 OBJETIVOS**

#### **GENERAL**

Proponer un modelo de integración de DevOps con metodologías ágiles basado en Scrum, incorporando métricas DORA y alineado con prácticas ITIL V4 para optimizar la entrega continua de servicios en NGS.

#### **ESPECÍFICOS**

1. Diseñar un plan estratégico de cierre de brechas en procesos, herramientas y competencias técnicas para el flujo CI/CD de NGS.
2. Desarrollar un modelo de integración CI/CD, incorporando prácticas DevOps y el marco de trabajo Scrum, incluyendo políticas operativas por fase del ciclo DevOps.
3. Alinear el modelo propuesto con las prácticas clave de ITIL V4 aplicables a la gestión de servicios tecnológicos en NGS.
4. Desarrollar un plan de integración de métricas DORA como parte del pipeline de entrega continua para una gestión basada en datos.

## **6.4 DESARROLLO DE LA PROPUESTA**

En esta sección se desarrolla de forma detallada la propuesta de aplicabilidad para NGS, describiendo las acciones y entregables que permiten cerrar las brechas identificadas en el diagnóstico del flujo CI/CD. A partir de los hallazgos del capítulo IV y las conclusiones del capítulo V, se estructura un plan de intervención orientado a estandarizar procesos, fortalecer competencias y consolidar el uso de herramientas y políticas que hagan posible una entrega

continua más consistente, trazable y controlada. Por ello, el desarrollo de la propuesta se presenta como un conjunto de componentes operativos que pueden adoptarse de manera progresiva, priorizando la viabilidad en el contexto actual de la organización y manteniendo la alineación con el marco ágil existente y con las buenas prácticas de gestión de servicios de ITIL v4.

#### 6.4.1 PLAN ESTRATÉGICO DE CIERRE DE BRECHAS PARA EL FLUJO CI/CD EN NGS

- **Estrategias de Mitigación de la Matriz FODA:** La matriz FODA constituye un instrumento fundamental para fortalecer el autoconocimiento organizacional de NGS y para identificar con precisión los factores internos y externos que influyen en la madurez de sus procesos, herramientas y competencias relacionadas con la adopción de prácticas DevOps y el flujo CI/CD. En el contexto de esta propuesta, el FODA permite orientar las acciones necesarias para reducir las brechas detectadas en el Capítulo IV, vinculándolas directamente con los resultados de la evaluación de madurez, los hallazgos de los instrumentos aplicados y las conclusiones establecidas en el Capítulo V.

Este análisis otorga una mayor coherencia estratégica a la propuesta, pues facilita priorizar intervenciones y diseñar respuestas alineadas tanto con las debilidades y amenazas identificadas como con las fortalezas y oportunidades que posee la organización. De esta manera, las estrategias de mitigación se convierten en el puente entre el diagnóstico y la transformación operativa que NGS requiere para avanzar hacia un modelo DevOps más estandarizado, automatizado y alineado con las mejores prácticas internacionales como ITIL V4.

Las estrategias derivadas de la matriz FODA se agrupan en cuatro categorías clave:

- **Estrategias FO (Fortalezas + Oportunidades):** Describen cómo aprovechar las fortalezas internas de NGS como su apertura al cambio y su base inicial en prácticas ITIL para potenciar las oportunidades externas que facilitan la adopción de DevOps y CI/CD.
- **Estrategias DO (Debilidades + Oportunidades):** Detallan cómo convertir las debilidades detectadas, como la falta de estandarización, escasa automatización y competencias técnicas limitadas, en mejoras sustanciales mediante la adopción de tecnologías, metodologías y herramientas emergentes.
- **Estrategias FA (Fortalezas + Amenazas):** Señalan de qué manera las fortalezas existentes pueden utilizarse para enfrentar riesgos externos, como la creciente demanda de

entregas rápidas, la presión competitiva y la necesidad de procesos más trazables y eficientes.

- **Estrategias DA (Debilidades + Amenazas):** Explican cómo mitigar las debilidades internas que, combinadas con factores externos adversos, podrían afectar la estabilidad operativa y la capacidad de NGS para sostener un modelo de entrega continua confiable.

**Tabla 46** Análisis y estrategias de mitigación de la Matriz FODA

		Análisis Interno	
		Fortalezas	Debilidades
		<ul style="list-style-type: none"> <li>Personal técnico capacitado en diversas tecnologías y con experiencia en entornos de desarrollo complejos.</li> <li>Uso de herramientas como Azure DevOps, sistemas de versionamiento y repositorios de código</li> <li>Experiencia previa en metodologías ágiles y prácticas DevOps.</li> <li>Uso de tableros Kanban para planificación y seguimiento.</li> </ul>	<ul style="list-style-type: none"> <li>Falta de documentación oficial de procesos operativos.</li> <li>Roles poco definidos dentro del marco ágil.</li> <li>Desaprovechamiento de programas internos de capacitación.</li> <li>Falta de KPIs para medir eficiencia, calidad y cumplimiento.</li> <li>Madurez insuficiente en herramientas de automatización y CI/CD.</li> </ul>
Estrategias	Oportunidades	Estrategias FO (Fortalezas + Oportunidades)	Estrategias DO (Debilidades + Oportunidades)
	<ul style="list-style-type: none"> <li>Existencia de programas internos de capacitación continua a través de plataformas educativas.</li> <li>Interés institucional en fortalecer la integración de prácticas DevOps-Ágil.</li> <li>Posibilidad de implementar herramientas de automatización como Docker y Kubernetes a nivel corporativo.</li> <li>Potencial para estandarizar procesos y roles mediante marcos ágiles e ITIL V4.</li> </ul>	<ul style="list-style-type: none"> <li>Liderar la estandarización de procesos utilizando el conocimiento técnico del equipo: Aprovechar la experiencia en Azure DevOps, Kanban y metodologías ágiles para formalizar flujos de trabajo y roles.</li> <li>Impulsar la adopción de herramientas de automatización, capitalizando la experiencia actual en DevOps para desplegar Docker, Kubernetes o pipelines.</li> <li>Fortalecer la integración DevOps-Ágil, usando los equipos con mayor madurez para pilotear la adopción formal del marco combinado.</li> <li>Aprovechar los programas internos de capacitación para consolidar el uso avanzado de repositorios, versionamiento y tableros Kanban.</li> </ul>	<ul style="list-style-type: none"> <li>Documentar procesos y roles apoyándose en programas de capacitación interna, reduciendo ambigüedades en responsabilidades y flujos de trabajo.</li> <li>Implementar gradualmente pipelines de CI/CD para disminuir la dependencia de procesos manuales y aumentar la confiabilidad de las entregas.</li> <li>Establecer KPIs para medir calidad, tiempos y eficiencia, utilizando herramientas disponibles en Azure DevOps y marcos de ITIL v4.</li> <li>Reducir la brecha de competencias técnicas incorporando un plan de formación estructurado en automatización, contenedores y gestión ágil.</li> </ul>
	Amenazas	Estrategias FA (Fortalezas + Amenazas)	Estrategias DA (Debilidades + Amenazas)

	<ul style="list-style-type: none"> <li>Resistencia al cambio por parte de algunos colaboradores ante la estandarización y automatización.</li> <li>Carga laboral y falta de tiempo disponible pueden limitar la innovación y adopción de nuevas prácticas.</li> <li>Riesgo de retrasos o errores en despliegues por falta de una arquitectura CI/CD consolidada.</li> <li>Incremento de la exigencia del entorno tecnológico y de los clientes respecto a la calidad y tiempos de entrega.</li> <li>La ausencia de procesos estandarizados puede afectar la trazabilidad y el control de versiones.</li> </ul>	<ul style="list-style-type: none"> <li>Usar personal experimentado como agentes de cambio.</li> <li>Aprovechar herramientas existentes para mejorar trazabilidad.</li> <li>Falta de estandarización afecta trazabilidad.</li> <li>Utilizar prácticas ágiles para enfrentar la presión del mercado.</li> <li>Capacitar al equipo para afrontar la evolución tecnológica.</li> </ul>	<ul style="list-style-type: none"> <li>Reducir tareas manuales mediante automatización.</li> <li>Documentar procesos para mejorar control y trazabilidad.</li> <li>Crear talleres de adopción para disminuir resistencia.</li> <li>Plan de capacitación escalonado para fortalecer CI/CD.</li> </ul>
--	--	--	--

Nota: Elaboración propia.

Para estas secciones se desarrolló un glosario con algunos términos clave que se usan en el transcurso del capítulo 6:

**Tabla 47** *Glosario de siglas y términos usados en el plan*

<b>Término</b>	<b>Significado</b>
<b>SM</b>	Scrum Master
<b>PO</b>	Product Owner
<b>PBI / PBIs</b>	Product Backlog Ítems
<b>DoD</b>	Definition of Done
<b>DoR</b>	Definition of Ready
<b>VCS</b>	Version Control System

Nota: Elaboración propia.

- **Matriz de Mitigación Técnica para Cierre de Brechas en Herramientas y Prácticas CI/CD:** Esta matriz tiene como finalidad proponer acciones de mejora que permitan escalar el nivel de madurez técnica de NGS en relación con sus prácticas DevOps actuales. A partir del diagnóstico inicial desarrollado en el capítulo IV, se definen intervenciones estratégicas para cada herramienta, política o práctica, con el fin de alcanzar un nivel alto de capacidad, asegurando sostenibilidad, eficiencia y control en el pipeline CI/CD.

**Tabla 48** *Matriz de Mitigación Técnica para Herramientas, Políticas y Prácticas en NGS*

No	Práctica / Herramienta	Nivel Actual	Nivel Esperado	Acciones de Mitigación	Responsable(s)	Indicador de Cumplimiento	Resultado Esperado
1	Azure DevOps para CI	Bajo	Alto	Implementar pipelines CI en Azure DevOps con validación automatizada en proyectos piloto.	Dev / QA	Azure DevOps integrado en el 100% de proyectos	Pipeline activo y estable
2	Uso de Docker	Medio	Alto	Capacitar en Docker avanzado y definir estándares de uso en despliegue.	RRHH / Devs	Contenedores implementados en entorno de desarrollo, pruebas y ambiente productivo.	Estandarización en despliegue
3	Git con buenas prácticas	Medio	Alto	Formalizar lineamientos para ramas, pull request, commits.	PM / Devs	Manual de Git documentado y aplicado	Gestión de código controlada
4	Políticas de merge y revisión	Bajo	Alto	Establecer políticas claras y revisiones obligatorias por pares.	QA / PM	90% de pull requests revisadas por pares	Mayor control de calidad
5	Pipeline de build	Bajo	Alto	Automatizar proceso de compilación con validaciones y alertas.	DevOps	Build sin fallas manuales recurrentes.	Compilación estable
6	Pruebas automatizadas	Bajo	Alto	Crear pruebas unitarias e integración con cobertura mayor al 70%.	QA / Devs	Cobertura de pruebas en cada build	Mayor fiabilidad de releases
7	Pipeline de despliegue	Bajo	Alto	Automatizar despliegue con verificación y rollback.	DevOps / QA	Porcentaje de despliegues exitosos mayor al 90%.	Entrega continua sin errores.

8	Rollback documentado	Medio	Alto	Formalizar y automatizar rollback en fallos.	QA / PM	Plan de rollback en todos los proyectos	Reducción de tiempo de recuperación
9	Monitoreo en CI/CD	Medio	Alto	Integrar dashboards (Grafana, App Insights) en etapas del pipeline.	QA / DevOps	Dashboard activo por cada entorno	Observabilidad total
10	Versionado	Medio	Alto	Aplicar versión semántica (semver) en releases.	PM / Devs	Releases con control de versión visible	Rastreo efectivo de cambios
11	Automatización de entornos	Medio	Alto	Aplicar IaC con Terraform o scripts ARM.	DevOps	Infraestructura aprovisionada automáticamente	Ambientes consistentes
12	Revisión continua	Medio	Alto	Revisión obligatoria de todos los PR por pares.	QA / Devs	100% de PR con doble aprobación	Mayor calidad de código
13	Infraestructura como código	Medio	Alto	Establecer repositorio de templates IaC por entorno.	DevOps / QA	80% de infraestructura gestionada como código	Escalabilidad y trazabilidad

Nota: Elaboración propia.

- Matriz de Evaluación y Priorización de Brechas:** A continuación del plan de mitigación propuesto, se presenta la Matriz de Criticidad de Brechas, la cual permite visualizar de forma estratégica el nivel de riesgo asociado a cada limitación identificada. Esta matriz considera dos dimensiones fundamentales: el impacto que cada brecha tiene sobre la operación del pipeline CI/CD y la probabilidad de que dicha brecha se manifieste o se mantenga en el tiempo si no se actúa. Su incorporación permite jerarquizar las intervenciones no solo desde una perspectiva técnica, sino también desde la gestión del riesgo, apoyando la toma de decisiones y el enfoque progresivo en la implementación del plan de mejora.

**Tabla 49** Matriz de criticidad para priorizar brechas con base al impacto y probabilidad

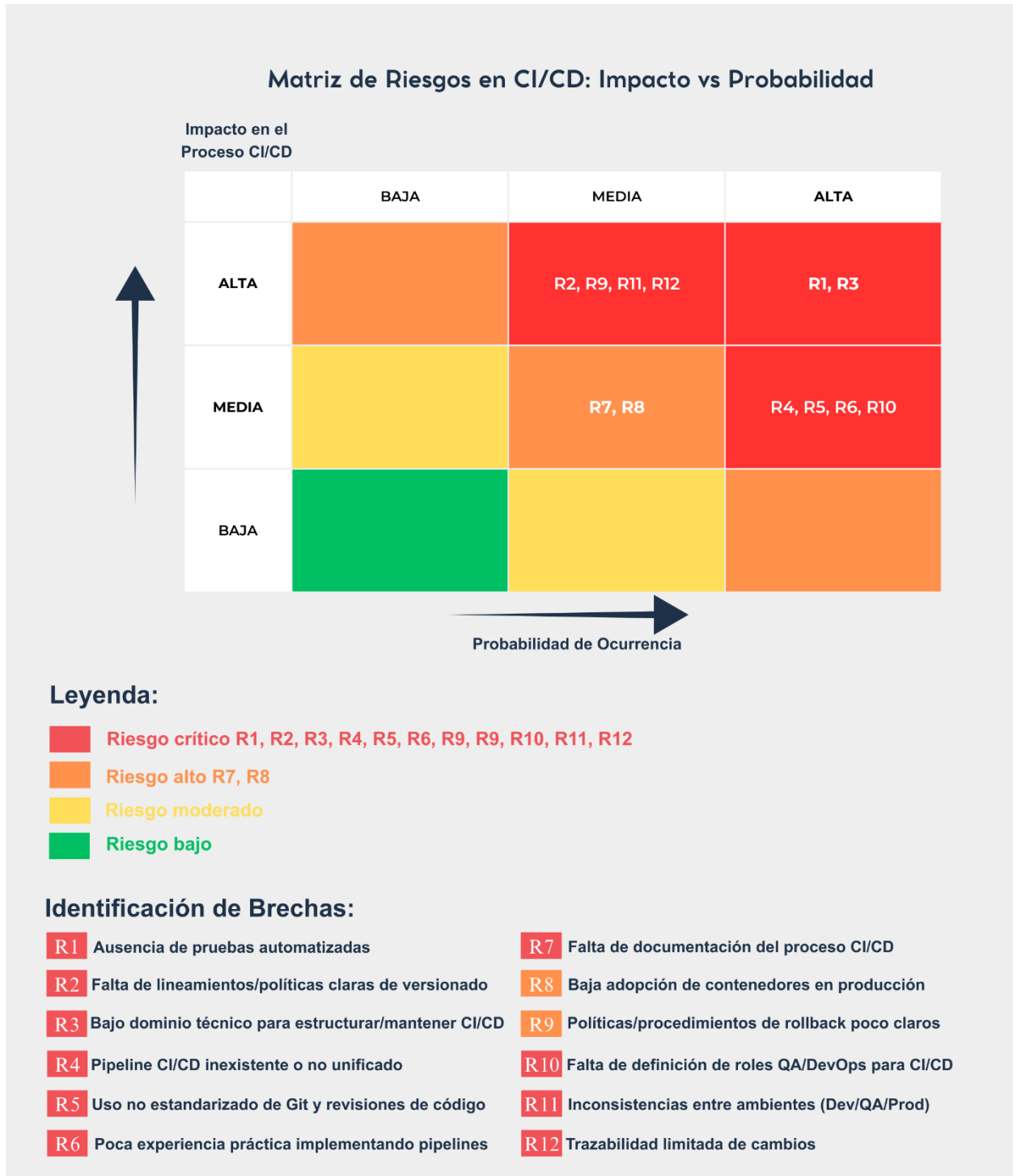
No	Brecha identificada	Categoría	Evidencia	Probabilidad	Impacto CI/CD	Criticidad	Justificación
1	Ausencia de pruebas automatizadas	Tecnología	Predominio de pruebas manuales y poca automatización.	Alta	Alta	<b>Crítica</b>	Reduce confiabilidad y velocidad de validación; aumenta defectos en etapas tardías.
2	Falta de lineamientos/políticas claras de versionado	Procesos	Ausencia de lineamientos claros para gestión de versiones; flujos/versiones informales.	Media	Alta	<b>Alta</b>	Afecta trazabilidad y control de releases; genera variabilidad entre equipos.
3	Bajo dominio técnico para estructurar/mantener CI/CD	Competencias	Competencias limitadas para pipelines consistentes; falta de habilidades para automatización.	Alta	Alta	<b>Crítica</b>	Bloquea estandarización y escalamiento del flujo automatizado.
4	Pipeline CI/CD inexistente o no unificado	Tecnología	Ausencia de pipeline unificado; actividades clave siguen siendo manuales.	Alta	Media	<b>Alta</b>	Incrementa errores y retrabajo; no garantiza repetibilidad del proceso.

5	Uso no estandarizado de Git y revisiones de código	Procesos	Procesos no estandarizados; variabilidad entre equipos y falta de lineamientos formales.	Alta	Media	<b>Alta</b>	Reduce control de integración; incrementa defectos y conflictos en merges.
6	Poca experiencia práctica implementando pipelines	Competencias	Conocimientos generales, aplicación limitada; habilidades insuficientes en automatización.	Alta	Media	<b>Alta</b>	Dificulta pasar de “pilotos” a adopción institucional (sostenibilidad).
7	Falta de documentación del proceso CI/CD	Procesos	Falta de documentación formal; dependencias personales y cuellos de botella.	Media	Media	<b>Media</b>	Aumenta dependencia del conocimiento tácito y reduce transferibilidad.
8	Baja adopción de contenedores en producción	Tecnología	Docker probado de forma limitada; falta de conocimiento práctico en contenedores.	Media	Media	<b>Media</b>	Limita ambientes reproducibles y portabilidad; afecta consistencia entre entornos.

9	Políticas/procedimientos de rollback poco claros	Procesos	Falta de puntos de control para detectar fallos temprano; afecta etapas críticas y estabilidad.	Media	Alta	<b>Alta</b>	Recuperación ante fallos no asegurada; mayor tiempo de restauración del servicio.
10	Falta de definición de roles QA/DevOps para CI/CD	Competencias	Coordinación ad hoc entre áreas (desarrollo, QA, infraestructura).	Alta	Media	<b>Alta</b>	Genera ambigüedad de responsabilidades y debilita la adopción de automatización.
11	Inconsistencias entre ambientes (Dev/QA/Prod)	Tecnología	Ambientes con inconsistencias que generan fallas repetitivas y retrabajos.	Media	Alta	<b>Alta</b>	Impide estabilidad del pipeline y provoca “funciona en mi máquina”.
12	Trazabilidad limitada de cambios	Procesos	Trazabilidad limitada; falta de gestión estandarizada de versiones.	Media	Alta	<b>Alta</b>	Dificulta auditoría, control de releases y análisis de incidentes.

Nota: Elaboración propia.

**Figura 101** Mapa de Calor de Riesgos - CI/CD



Nota: Elaboración propia.

El mapa de calor evidencia que las brechas con mayor criticidad se concentran en la falta de automatización del pipeline y en las capacidades técnicas necesarias para operarlo de forma

consistente, junto con debilidades de gobernanza como versionado, rollback, trazabilidad y definición de responsabilidades. En consecuencia, la prioridad no es incorporar más herramientas, sino estandarizar su uso, habilitar prácticas mínimas verificables (plantillas de pipeline, políticas de repositorio) y fortalecer habilidades prácticas mediante un plan de capacitación aplicado. Con base en esta priorización, el siguiente apartado presenta el plan de acciones para el cierre de brechas, orientado a implementar mejoras progresivas y medibles que reduzcan el riesgo operativo y aumenten la repetibilidad del flujo de entrega continua.

- **Plan de acciones para el cierre de brechas CI/CD en NGS:** Con base en la priorización obtenida en la matriz de criticidad y su mapa de calor, se definió un plan de acciones orientado a cerrar brechas de alto impacto mediante intervenciones prácticas y verificables. El enfoque del plan se concentra en dos frentes:
  1. Estandarizar el uso de herramientas clave del ciclo CI/CD (repositorios, pipelines, políticas de calidad, trazabilidad, documentación y mecanismos de recuperación)
  2. Fortalecer capacidades técnicas con acompañamiento aplicado a proyectos piloto.

El resultado esperado es reducir el riesgo operativo, aumentar la repetibilidad del despliegue y establecer una base institucional mínima para escalar la entrega continua en NGS.

**Figura 102** Plan de Acción para optimizar CI/CD



Nota: Elaboración propia.

La figura “Plan de acción para optimizar CI/CD” resume en tres componentes las acciones que se detallan en la matriz a continuación: Estandarización Técnica (R4: plantilla estándar de pipeline; R2 y R5: políticas de versionado, PR y control de cambios; R9: runbooks y rollback), pilotos guiados (R6: proyectos piloto con mentoría semanal y validación de despliegue para generar evidencia operativa) y capacitación aplicada (R3 y R6: fortalecimiento práctico en CI/CD; R8 y R11: uso de Docker y consistencia de ambientes; R12: trazabilidad del cambio). En conjunto, estos tres bloques conectan la priorización del mapa de calor con acciones ejecutables y verificables, que se presentan en la siguiente matriz del plan de acción.

**Tabla 50** Matriz del plan de acción (acciones, entregables y evidencia)

R	Brecha priorizada	Acción del plan	Herramienta / Entregable	Responsable(s)	Plazo	Indicador / evidencia de cierre
R4	Pipeline no unificado	Diseñar e institucionalizar una plantilla estándar de pipeline.	Azure DevOps Pipelines (template YAML) + guía de uso	DevOps + Dev Lead	2 semanas	2 proyectos piloto ejecutando template; ≥80% proyectos adoptan template al cierre.
R5	Uso no estandarizado de Git/PR.	Implementar políticas de repositorio: PR obligatorio, revisor, Checklist, build requerido.	Policiés Azure Repos + checklist PR	QA Lead + Dev Lead	2 semanas	100% merges por PR; build requerido activo.
R2	Versionado sin estándar.	Definir reglas de versionado, ramas, tags y releases.	Guía de versionado + convención de ramas	PM + Dev Lead	2-3 semanas	Releases con tags; convención aplicada en pilotos.
R12	Trazabilidad limitada	Estandarizar trazabilidad: ticket - commit - PR - release.	Convención commits/branches + vínculo work items.	PM + Devs	3 semanas	≥80% PR vinculados a ticket; releases trazables.
R1	Sin pruebas automatizadas	Implementar pruebas mínimas (unitarias + smoke) e integrarlas al pipeline.	Framework de tests + paso “test” en pipeline.	QA + Devs	4-6 semanas	Reporte automático de pruebas en build; fallas bloquean merge (gate)
R6	Poca práctica en pipelines	Ejecutar pilotos guiados y mentoría semanal aplicada.	2 proyectos piloto + bitácora de mejoras	DevOps + QA	4-6 semanas	Pipeline estable y repetible en pilotos; lecciones aprendidas documentadas

R9	Rollback poco claro	Definir procedimiento de rollback y probarlo en piloto	Runbook de rollback + procedimiento validado	DevOps + QA	4-6 semanas	Simulación de rollback realizada y documentada
R7	Falta de documentación	Implementar “documentación mínima CI/CD” por proyecto	Wiki/Runbooks: pipeline, despliegue, rollback, accesos	QA + DevOps	6-8 semanas	Wiki completa en pilotos; checklist de documentación aprobado
R11	Ambientes inconsistentes	Normalizar Dev/QA usando plantillas/scripts y/o contenedores base	Scripts de ambiente + Checklist consistencia	DevOps	7-10 semanas	Dev/QA reproducibles; reducción de incidencias “funciona en mi máquina”
R8	Baja adopción de contenedores	Adopción gradual: contenedor en Dev/QA para pilotos	Dockerfile + imágenes base + guía Docker	DevOps + Devs	7-10 semanas	2 pilotos desplegados en Dev/QA con contenedores
R3	Bajo dominio técnico CI/CD	Capacitación aplicada (taller + laboratorios) sobre Git, pipelines, tests, Docker	Plan de formación (O1-E5) + evaluación	RRHH + DevOps + QA	8 semanas	≥80% aprobación laboratorios; entregables por módulo
R10	Roles no definidos	Definir responsables mínimos CI/CD (propietario pipeline, QA gatekeeper, reviewers)	RACI básico CI/CD (sin entrar a Scrum aún)	Gerencia TI + RRHH	2 semanas	RACI aprobado y comunicado; responsables asignados por proyecto

Nota: Elaboración propia.

Para garantizar que el plan produzca resultados verificables, se propone un esquema de seguimiento basado en indicadores de adopción y cumplimiento, monitoreados de forma semanal o quincenal durante la ejecución del plan.

**Tabla 51** *Seguimiento y control del plan de acción*

Indicador	Cómo se mide	Frecuencia	Meta de cumplimiento	Responsable
Repos con PR obligatorio + build requerido	Repos con políticas activas / total repos	Semanal	≥80%	QA Lead
Proyectos con template de pipeline	Proyectos con template / total	Semanal	≥80%	DevOps
Builds con pruebas automáticas	Builds con paso de tests / total builds	Semanal	≥70% en pilotos	QA Lead
Proyectos con runbook de rollback	runbooks existentes / total	Quincenal	≥80%	DevOps
PR con trazabilidad ticket-PR-release	PR vinculados / total PR	Semanal	≥80%	PM
Documentación mínima completada	Proyectos con wiki CI/CD / total	Quincenal	≥80%	QA Lead
Cumplimiento del cronograma	Actividades completadas / planificadas	Semanal	≥85%	PM

Nota: Elaboración propia.

El plan de acción define intervenciones técnicas y organizativas que permiten cerrar brechas críticas del flujo CI/CD con entregables verificables y control de avance. Para asegurar sostenibilidad y reducir la dependencia de conocimiento tácito, el cierre se complementa con un plan de capacitación aplicada y evaluación de competencias, el cual se presenta en el siguiente entregable.

- **Plan de formación técnica para cierre de brechas de competencias CI/CD en NGS:** Con el fin de reducir brechas de competencias asociadas a CI/CD y asegurar sostenibilidad de la mejora, se define un plan general institucional de fortalecimiento de capacidades. Este plan establece una ruta de aprendizaje continua, mecanismos de evaluación y criterios de

evidencia técnica, de modo que la capacitación se refleje en adopción real de herramientas y controles del pipeline (políticas de repositorio, pipelines estandarizados, pruebas automatizadas, contenedores, despliegue y trazabilidad).

Para acelerar el cierre de brechas críticas de competencias, se propone ejecutar un plan intensivo con un grupo selecto de desarrolladores y QA (equipo núcleo), facilitado por un Ingeniero DevOps con experiencia en YAML, Azure DevOps Pipelines, Docker y Kubernetes. El objetivo es que el grupo núcleo produzca entregables reales sobre proyectos piloto y actúe como multiplicador interno, replicando la práctica con el resto de los desarrolladores mediante sesiones de transferencia y acompañamiento.

Se conformará un grupo núcleo para acelerar el cierre de brechas de competencias asociadas a CI/CD. Este grupo estará integrado por 6 a 8 personas, incluyendo 4 desarrolladores, 2 integrantes de QA y 1 líder técnico (con la posibilidad de incorporar 1 apoyo de infraestructura si aplica). La selección se realizará con base en dos criterios principales: participación activa en proyectos en ejecución y disposición para recibir mentoría y replicar el conocimiento hacia el resto del equipo, con el objetivo de facilitar una expansión progresiva de las capacidades técnicas dentro de NGS.

**Tabla 52** Talleres para grupo selecto de desarrolladores

Semana	Taller (DevOps Engineer)	Objetivo del taller	Laboratorio obligatorio (evidencia)	Brechas que cierra
1	Git avanzado + PR policies + versionado	Controlar integración y releases	Repo con PR obligatorio + build requerido + tags	R2, R5, R12
2	YAML + pipeline template (Azure DevOps)	Pipeline estandarizado	Template YAML funcionando en 2 repos piloto	R4, R6
3	Quality gates	Asegurar calidad antes del merge	PR bloqueado si falla pipeline + checklist PR	R5, R4
4	Automatización de pruebas (QA Automation)	Reducir defectos y acelerar validación	Tests unit/smoke corriendo en pipeline + reporte	R1

5	Docker aplicado (Dev/QA)	Ambientes reproducibles	Imagen base + ejecución en Dev/QA	R8, R11
6	Kubernetes operativo + despliegue controlado	Despliegue y recuperación	Deploy controlado + procedimiento rollback	R9, R11
7	Trazabilidad end-to-end	Auditoría y control del cambio	Ticket-commit-PR-release en 1 ciclo	R12
8	Documentación mínima + runbooks	Sostenibilidad y transferencia	Wiki CI/CD + runbook rollback + despliegue	R7, R10

Nota: Elaboración propia.

Una vez consolidado el equipo núcleo, la adopción se expande mediante un esquema de transferencia interna, donde el grupo selecto replican prácticas y acompañan la implementación en nuevos proyectos utilizando plantillas, guías y checklist creados durante el taller intensivo.

**Tabla 53** *Expansión y transferencia liderada por el Grupo Núcleo*

Actividad	Responsable	Alcance	Evidencia
Sesiones de réplica (2 por tema)	Grupo Núcleo + QA Lead	Todo el equipo	Lista de asistencia + mini-lab aprobado
Implementación asistida en proyectos no piloto	Grupo Núcleo + DevOps (soporte puntual)	3-5 proyectos	Pipeline template adoptado + políticas activas
Revisión semanal de adopción (auditoría)	PM + QA Lead + Grupo Núcleo	Todos	Reporte de cumplimiento con acciones correctivas
Documentación institucional final	Grupo Núcleo + QA + DevOps	Todos	Guías publicadas + repositorio de plantillas

Nota: Elaboración propia.

El plan de expansión consolida la transferencia de conocimiento mediante un esquema liderado por el Grupo Núcleo, quien replica los temas clave y acompaña la adopción en proyectos no piloto hasta lograr que las prácticas queden instaladas en el trabajo diario. Este enfoque permite escalar de forma controlada lo implementado en los pilotos, asegurando consistencia en el uso de

plantillas, políticas y estándares técnicos, y reduciendo la dependencia de apoyo externo a medida que la capacidad interna se fortalece.

Se propone un checklist de adopción CI/CD como instrumento inicial para validación interna de NGS, con el fin de comprobar por proyecto la evidencia mínima de implementación (políticas, pipeline estándar, pruebas, trazabilidad, documentación y rollback) y facilitar auditorías periódicas.

**Tabla 54** Checklist propuesto de adopción CI/CD por proyecto

Ítem de verificación	Sí/No	Evidencia requerida	Justificación (por qué / observaciones)
PR obligatorio activo (rama principal)		Configuración visible en repositorio	
Build requerido antes de merge		Política de repositorio + ejecución exitosa	
Template estándar de pipeline implementado		YAML/template referenciado en el proyecto	
Pipeline genera artefacto versionado		Artefactos/releases con versión	
Convención de versionado aplicada		Tags/releases consistentes	
Trazabilidad ticket-commit-PR-release		Link en PR y en release	
Pruebas automáticas ejecutándose en pipeline		Reporte de pruebas en build	
Quality gate (bloquea merge si falla)		Evidencia de bloqueo por falla	
Runbook de despliegue y rollback disponible		Documento publicado + prueba realizada	
Documentación mínima CI/CD publicada		Wiki/runbook por proyecto	
Ambientes Dev/QA consistentes (checklist)		Evidencia reproducible / checklist firmado	
Docker implementado en Dev/QA (si aplica)		Dockerfile/imagen base + ejecución	
Roles mínimos CI/CD definidos (RACI)		Documento publicado y comunicado	

Nota: Elaboración propia.

**Plan de mitigación para elevar la madurez de las prácticas ITIL V4 evaluadas en NGS:** A partir de la matriz consolidada de madurez desarrollada en el capítulo de resultados, se propone una matriz de mitigación orientada a cerrar brechas de manera gradual y realista. Esta matriz traduce los hallazgos de madurez por práctica ITIL v4 en acciones operativas apoyadas por prácticas DevOps ya identificadas para NGS, con responsables claros y evidencias observables. El propósito no es medir con métricas numéricas, sino asegurar consistencia, estandarización y sostenibilidad en la gestión del servicio mientras se fortalece el flujo CI/CD.

**Tabla 55** *Matriz de mitigación para subir madurez ITIL v4 apoyándose en DevOps*

Práctica ITIL v4	Brecha actual	Objetivo de mejora	Acciones de mitigación propuestas	Prácticas DevOps que habilitan	Responsables	Evidencia observable
Gestión de cambios	Leve	Uniformar el control de cambios entre proyectos	Estandarizar “flujo mínimo de cambio” ligado al repositorio (PR + trazabilidad), aprobación definida y registro simple	VCS, Gestión colaborativa del flujo, CI	PO/PM + Líder técnico + QA Lead	PR con referencia a PBI, registro de aprobación, bitácora de cambios
Gestión de despliegues	Moderada	Pasar de despliegues disparejos a un proceso repetible	Plantilla de pipeline y pasos mínimos por ambiente; runbook de despliegue/rollback; validación post-release	CD, CI, Automatización de pruebas	DevOps + QA Lead + Dev Team	Template adoptado, bitácora de despliegues, evidencia de rollback probado
Gestión de incidentes	Leve	Estandarizar registro y respuesta, reduciendo reactividad	Ruta de escalamiento, registro consistente y retroalimentación al backlog desde incidentes	Monitoreo y retroalimentación, Gestión colaborativa del flujo	QA Lead + DevOps + SM	Registro de incidentes, acciones correctivas en backlog, notas de revisión post-incidente

Gestión de problemas	Leve-moderada	Fortalecer análisis de causa raíz y recurrencia	Convertir fallas repetidas del pipeline/incidentes en PBIs técnicos; práctica básica de RCA en retrospectiva técnica	Automatización de pruebas, Monitoreo y retroalimentación, CI	QA Lead + Líder técnico + SM	PBIs técnicos recurrentes, minutas de retro técnica, lecciones aprendidas documentadas
Mejora continua	Moderada-alta	Instalar un ciclo de mejora estable y repetible	Retrospectiva técnica con acciones obligatorias; revisión periódica del flujo de valor; priorización de mejoras en backlog	Gestión colaborativa del flujo, Monitoreo y retroalimentación	SM + PO + Equipo núcleo	Acciones de mejora registradas, evidencia de seguimiento, ajustes a guías/plantillas
Monitoreo y gestión de eventos	Leve-moderada	Aumentar visibilidad y alertamiento básico post-release	Monitoreo mínimo por proyecto (logs/alertas/métricas básicas) + revisión post-despliegue; dashboard sencillo	Monitoreo y retroalimentación, CD	DevOps + Dev Team	Dashboard/bitácora, alertas configuradas, reporte breve post-release
Gestión de configuración y activos	Moderada-alta	Reducir “desconocimiento” y dispersión de configuraciones	Inventario mínimo y control por repositorio; estandarizar configuración de ambientes con plantillas/scripts	IaC, VCS, CD	DevOps + Infra (si aplica) + Líder técnico	Repositorio con configuración, inventario mínimo, consistencia entre entornos

Nota: Elaboración propia.

Esta matriz permite convertir la brecha de madurez identificada por práctica ITIL v4 en un conjunto de acciones aplicables, apoyadas por prácticas DevOps que facilitan estandarización, trazabilidad y retroalimentación continua. Al mantener responsables y evidencias observables, la organización puede fortalecer la gestión del servicio sin recargar el modelo con mediciones complejas. A partir de estas acciones, el cronograma de implementación organiza la ejecución progresiva por fases y facilita su adopción sostenida en proyectos piloto y su posterior expansión.

- **Propuesta de creación del Área DevOps en la consultora Nueva Generación de Software (NGS):** Se plantea como propuesta organizacional la posibilidad de que NGS, de manera progresiva y conforme a su nivel de madurez y prioridades estratégicas, considere la creación de un Área DevOps como un mecanismo para consolidar, gobernar y optimizar el flujo CI/CD. Esta propuesta no implica una implementación inmediata ni una reestructuración obligatoria, sino que se concibe como una alternativa evolutiva que puede ser adoptada gradualmente, apoyándose inicialmente en esquemas de consultoría especializada y fortaleciendo la capacidad interna de la organización en el tiempo.

El objetivo es establecer un área DevOps que permita diseñar, gobernar y optimizar el flujo de integración y entrega continua (CI/CD) en NGS, asegurando la automatización, la estandarización de procesos, la medición del desempeño mediante métricas y la confiabilidad del servicio tecnológico.

El Área DevOps tendrá como alcance:

- La definición y mantenimiento de políticas DevOps (CI/CD, control de versiones, calidad y seguridad).
- La automatización de procesos de integración, pruebas y despliegue.
- La gestión de la confiabilidad y estabilidad de los servicios tecnológicos.
- La implementación y seguimiento de métricas de desempeño (DORA).
- El soporte transversal a los equipos Scrum y al área de operaciones.
- La alineación del ciclo de entrega de software con la gestión del servicio bajo ITIL v4.

Con el fin de ofrecer una referencia organizacional para la adopción progresiva del modelo DevOps propuesto, se presenta la ficha técnica del puesto de Ingeniero DevOps como un perfil orientativo, que podrá ser considerado por NGS en el mediano plazo, ya sea para el fortalecimiento de capacidades internas o como base para procesos de consultoría especializada.

**Tabla 56** *Ficha técnica del puesto DevOps*

<b>Puesto</b>	Ingeniero DevOps
<b>Objetivo General</b>	Diseñar, implementar y mantener soluciones DevOps orientadas a la automatización de infraestructura y pipelines CI/CD, garantizando la entrega continua de software, la estabilidad operativa, la escalabilidad de los servicios y la mejora continua de los procesos tecnológicos de NGS.
<b>Funciones Principales</b>	Diseñar, implementar y mantener pipelines de integración continua y entrega continua (CI/CD), Automatizar procesos de build, pruebas, despliegue y rollback en ambientes de desarrollo, QA y producción, Administrar infraestructura como código, contenedores y plataformas de orquestación, Implementar monitoreo, observabilidad, alertas, escalabilidad automática y recuperación ante fallos, Integrar prácticas DevOps con equipos de desarrollo, QA, arquitectura y seguridad, Aplicar controles de seguridad (DevSecOps) en pipelines e infraestructura, Participar como primera o segunda línea en la gestión de incidentes productivos, Proponer mejoras técnicas basadas en métricas y análisis de desempeño.
<b>Nivel Educativo</b>	Egresado o Graduado de Ingeniería o Licenciatura en Sistemas, Informática o carreras afines.
<b>Conocimientos Especializados</b>	Administración de sistemas Linux / Unix, Gestión de infraestructura en la nube (AWS, Azure o Google Cloud Platform), Herramientas CI/CD (Azure DevOps, GitLab CI/CD o similares). Contenedores y orquestación (Docker y Kubernetes). Experiencia en aplicaciones Java, .NET o NodeJS (Spring / Spring Boot deseable). Monitoreo y observabilidad con New Relic o Dynatrace. Infraestructura como código y automatización. Metodologías ágiles y enfoque DevOps.
<b>Idiomas</b>	Inglés: Nivel intermedio.
<b>Experiencia Laboral</b>	Experiencia comprobable en entornos DevOps, automatización de infraestructura, gestión de pipelines CI/CD y soporte a ambientes productivos.
<b>Competencias</b>	Alto compromiso con la calidad y la excelencia, Capacidad de análisis y resolución de problemas, Comunicación efectiva y trabajo en equipo, tolerancia al estrés en ambientes productivos, pensamiento crítico y criterio propio, proactividad, responsabilidad y compromiso, Capacidad autodidacta y orientación a la innovación, Planificación y organización.
<b>Relaciones Externas</b>	Proveedores de tecnología y servicios cloud.
<b>Relaciones Internas</b>	Infraestructura, Desarrollo, QA, Arquitectura, Seguridad, SCM / DevOps IT.

<b>Equipo / Herramientas de Trabajo</b>	Laptop / Computadora, teléfono móvil, herramientas colaborativas (Teams), plataformas CI/CD, cloud y Monitoreo.
<b>Valor Estratégico del Puesto</b>	El Ingeniero DevOps contribuye directamente a elevar la madurez DevOps de NGS, optimizando la entrega continua, reduciendo riesgos operativos, mejorando la estabilidad de los servicios y consolidando una cultura de automatización, colaboración y mejora continua alineada con la fábrica digital.

Nota: Elaboración propia.

La ficha técnica del puesto de Ingeniero DevOps presentada permite establecer una referencia clara sobre las competencias, responsabilidades y el valor estratégico asociado a este rol dentro de un contexto de adopción progresiva del modelo DevOps. A partir de este perfil orientativo, se amplía a continuación la propuesta mediante la identificación de las áreas y roles asociados al Ingeniero DevOps, considerando su interacción con otros perfiles especializados y su alineación con prácticas DevOps, ITIL v4 y Scrum, tal como se detalla a continuación:

**Tabla 57** Areas o roles del ingeniero DevOps

Área / Rol	Objetivo del rol	Funciones y responsabilidades	Alineación metodológica (DevOps - ITIL v4 - Scrum)	Beneficio para NGS
<b>Área DevOps</b>	Estandarizar, gobernar y optimizar el flujo de integración y entrega continua (CI/CD)	Definir políticas DevOps (CI/CD, control de versiones, calidad y seguridad), Diseñar y mantener pipelines CI/CD, Integrar automatización de pruebas y despliegues, Implementar métricas de desempeño (DORA), Brindar soporte transversal a equipos Scrum y Operaciones	DevOps: automatización y entrega continua, ITIL v4: gestión de cambios, versiones y mejora continua, Scrum: soporte al ciclo del Sprint	Reducción de tiempos de entrega, mayor calidad del software y estandarización organizacional
<b>Líder DevOps / Platform Lead</b>	Dirigir estratégicamente la adopción DevOps en la organización	Definir la estrategia DevOps alineada al negocio, Coordinar con el CEO y la alta dirección, Gobernar políticas y estándares técnicos, Supervisar métricas DORA y KPIs de servicio, Liderar la mejora continua del modelo	DevOps: liderazgo técnico y cultural, ITIL v4: gobernanza y gestión del valor, Scrum: alineación estratégica	Asegura sostenibilidad del modelo, control y alineación estratégica

<b>Ingeniero DevOps</b>	Implementar y operar la automatización del flujo CI/CD	Construir y mantener pipelines CI/CD, Automatizar pruebas, despliegues e infraestructura (IaC), Integrar herramientas de control de versiones, calidad y seguridad, Documentar estándares técnicos	DevOps: CI/CD y automatización, ITIL v4: gestión de versiones y despliegues, Scrum: soporte técnico al desarrollo	Incrementa frecuencia de despliegues, reduce errores y retrabajos
<b>Ingeniero Site Reliability Engineering (SRE)</b>	Garantizar confiabilidad, estabilidad y disponibilidad de los servicios	Definir y monitorear SLI, SLO y SLA, Gestionar incidentes y ejecutar postmortems, Reducir el tiempo medio de recuperación (MTTR), Automatizar tareas operativas repetitivas	SRE: confiabilidad y resiliencia, ITIL v4: gestión de incidentes, continuidad y disponibilidad, DevOps: operación automatizada	Mejora la estabilidad del servicio, reduce interrupciones y fortalece la confianza del cliente
<b>Relación transversal con equipos Scrum</b>	Facilitar colaboración y alineación operativa	Acompañar releases y despliegues, Proveer retroalimentación basada en métricas, Apoyar la mejora continua del proceso	Scrum: trabajo colaborativo, DevOps: integración Dev-Ops	Mayor previsibilidad y eficiencia en la entrega de software

Nota: Elaboración propia.

La propuesta de creación del Área DevOps y la definición de roles asociados se plantea como una alternativa organizacional de carácter evolutivo, orientada a fortalecer progresivamente la madurez DevOps de NGS y a consolidar las prácticas de integración y entrega continua descritas en el modelo propuesto. Este planteamiento no supone una reestructuración inmediata ni una adopción obligatoria, sino que ofrece a la organización un marco de referencia que puede ser analizado y considerado conforme a sus prioridades estratégicas, capacidades internas y resultados obtenidos durante las etapas iniciales de capacitación y acompañamiento especializado.

En este sentido, la propuesta reconoce que NGS puede optar inicialmente por esquemas de consultoría y transferencia de conocimiento con un Ingeniero DevOps, permitiendo preparar a los equipos y validar el enfoque antes de avanzar hacia una estructura interna formal. De esta manera, la eventual conformación de un Área DevOps se concibe como un paso posterior y opcional, alineado con un proceso de mejora continua, crecimiento organizacional y sostenibilidad del modelo DevOps-Ágil-ITIL v4 propuesto.

Luego de definir y ejecutar el plan de acciones y formación para el cierre de brechas priorizadas, se establece una base técnica y de capacidades que habilita el siguiente paso de la propuesta. En este contexto, se plantea el diseño de un modelo de integración CI/CD que incorpore prácticas DevOps para estandarizar y automatizar el ciclo de entrega, y que se integre de forma coherente con el marco ágil actual basado en Scrum, asegurando alineación entre la ejecución técnica, la gestión del trabajo y los mecanismos de control y calidad.

#### 6.4.2 MODELO DE INTEGRACIÓN CI/CD INCORPORACIÓN DE PRÁCTICAS DEVOPS AL MARCO SCRUM Y ALINEACIÓN CON BUENAS PRÁCTICAS DE ITIL V4

En este objetivo se presenta un modelo de integración CI/CD para NGS orientado a implementar las prácticas DevOps identificadas y operarlas dentro del marco ágil Scrum ya utilizado por la organización, asegurando coherencia entre la gestión del trabajo y la ejecución técnica del ciclo de entrega. La propuesta se fundamenta en los hallazgos del Capítulo IV, donde se evidenció la necesidad de estandarizar prácticas, herramientas y criterios operativos para reducir variabilidad entre proyectos, fortalecer automatización y mejorar trazabilidad y control. Asimismo, el modelo incorpora la alineación con las prácticas ITIL v4 seleccionadas, de manera que la entrega continua se ejecute con gobernanza ligera en cambios, despliegues, monitoreo y mejora continua, sin afectar la agilidad del equipo.

A continuación se desarrolla cada plan orientado a la integración de prácticas y políticas DevOps:

- **Plan de implementación de prácticas DevOps en NGS:** En NGS se identificaron prácticas DevOps con alto potencial de adopción, pero con aplicación no uniforme entre proyectos y dependencia de esfuerzos manuales. Este plan se desarrolla para convertir esas prácticas en acciones implementables dentro del trabajo diario, asegurando consistencia, trazabilidad y control, sin alterar la dinámica ágil basada en Scrum.

El plan funciona como una guía operativa ya que define qué práctica se implementa, los pasos concretos para ejecutarla, cómo se integra a los eventos y artefactos Scrum, cómo se alinea a las prácticas ITIL v4 seleccionadas, quién la lidera y qué evidencia mínima debe quedar. De esta forma, la adopción se vuelve institucional y verificable.

**Tabla 58** Matriz de implementación de prácticas DevOps identificadas para NGS

Práctica DevOps (NGS)	Propósito	¿Cómo se implementa?	Integración con Scrum	Alineación ITIL v4 (7 prácticas)	Responsable(s)	Evidencia mínima
<b>1. Integración continua (CI)</b>	Validar cambios continuamente y reducir integración tardía	<ol style="list-style-type: none"> <li>1. PR obligatorio + revisión.</li> <li>2. Build automático por PR/commit.</li> <li>3. Validaciones automáticas visibles.</li> </ol>	<ul style="list-style-type: none"> <li>• Durante el Sprint.</li> <li>• En Daily se gestionan fallos como impedimentos.</li> </ul>	Gestión de cambios; Gestión de Configuración y Activos; Mejora continua	Dev Team + DevOps (soporte) + Líder técnico	PR revisado + registro de build/validaciones
<b>2. Entrega continua (CD)</b>	Preparar entregas repetibles y controladas	<ol style="list-style-type: none"> <li>1. Artefacto versionado por entrega.</li> <li>2. Promoción a Dev/QA con registro.</li> <li>3. Producción con aprobación cuando aplique + rollback definido.</li> </ol>	<ul style="list-style-type: none"> <li>• Review puede demostrarse desde ambiente controlado.</li> <li>• El incremento debe ser “entregable” según DoD técnico.</li> </ul>	Gestión de despliegues; Gestión de cambios; Gestión de incidentes	DevOps + QA Lead + Dev Team	Release versionado + registro de despliegue + evidencia de aprobación
<b>3. Control de versiones (VCS)</b>	Trazabilidad y control del código	<ol style="list-style-type: none"> <li>1. Política de ramas/merges.</li> <li>2. Convención de versionado/tags.</li> <li>3. Ticket-commit-PR obligatorio.</li> </ol>	<ul style="list-style-type: none"> <li>• Planning exige trazabilidad por historia.</li> <li>• Daily gestiona bloqueos por merge.</li> </ul>	Gestión de cambios; Gestión de Configuración y Activos	Dev Team + Líder técnico	Política publicada + tags consistentes + trazabilidad completa

<p><b>4. Infraestructura como código (IaC)</b></p>	<p>Reducir la variabilidad de ambientes y mejorar la repetibilidad.</p>	<ol style="list-style-type: none"> <li>Plantillas para variables/ambientes.</li> <li>Versionar scripts/plantillas.</li> <li>Ejecución controlada para aprovisionar/configurar.</li> </ol>	<ul style="list-style-type: none"> <li>Se gestiona como backlog técnico.</li> <li>Retro prioriza mejoras.</li> </ul>	<p>Gestión de Configuración y Activos; Gestión de cambios; Gestión de despliegues</p>	<p>DevOps + (Infra si aplica) + Líder técnico</p>	<p>Plantillas/scripts versionados + evidencia de ejecución</p>
<p><b>5. Automatización de pruebas</b></p>	<p>Reducir pruebas manuales y elevar confiabilidad</p>	<ol style="list-style-type: none"> <li>Set mínimo (unit/smoke).</li> <li>Integración al pipeline.</li> <li>Reporte automático + fallos repetitivos a mejora.</li> </ol>	<ul style="list-style-type: none"> <li>DoD exige evidencia.</li> <li>Review respalda con evidencias cuando aplique.</li> </ul>	<p>Gestión de problemas; Problemas; Mejora continua</p>	<p>QA/QA Lead + Dev Team</p>	<p>Reporte automático + criterios mínimos + registro de fallos</p>
<p><b>6. Monitoreo y retroalimentación</b></p>	<p>Mejorar con señales post-entrega</p>	<ol style="list-style-type: none"> <li>Señales mínimas (logs/alertas/métricas).</li> <li>Revisión post-entrega.</li> <li>Hallazgos a PBIs técnicos.</li> </ol>	<ul style="list-style-type: none"> <li>Daily visualiza fallos; Retro define mejoras; se alimenta el backlog.</li> </ul>	<p>Monitoreo y Gestión de Eventos; Gestión de incidentes; Gestión de problemas; Mejora continua</p>	<p>Dev Team + QA + (Ops/DevOps)</p>	<p>Panel básico + bitácora de eventos + PBIs creados</p>
<p><b>7. Gestión colaborativa del flujo</b></p>	<p>Coordinar Dev-QA-DevOps y hacer visible el trabajo</p>	<ol style="list-style-type: none"> <li>Roles claros (RACI).</li> <li>Tablero actualizado + reglas de trabajo.</li> <li>Acompañamiento cruzado (shift-left).</li> </ol>	<ul style="list-style-type: none"> <li>Scrum completo: Planning/Daily/Review/Retro.</li> </ul>	<p>Mejora continua; Gestión de cambios</p>	<p>SM + Líder técnico + Equipo</p>	<p>RACI + acuerdos de trabajo + tablero actualizado</p>

Nota: Elaboración propia.

Con la matriz de implementación se deja establecido un marco práctico para llevar las prácticas DevOps identificadas en NGS desde un nivel conceptual a un nivel operativo, definiendo qué debe ejecutarse, quién lo lidera y qué evidencias mínimas deben generarse. Esto permite que la adopción no dependa de iniciativas aisladas, sino que se convierta en un estándar replicable entre proyectos, manteniendo coherencia con la dinámica Scrum y fortaleciendo la trazabilidad del trabajo técnico.

A partir de esta base, el siguiente paso consiste en ordenar la adopción mediante un plan gradual y, posteriormente, formalizar políticas operativas que aseguren consistencia por fase del ciclo DevOps. De esta forma, la implementación puede avanzarse de manera progresiva, medible y sostenible, facilitando su validación por parte de NGS y su expansión institucional.

- Plan de integración por etapas para la implementación gradual de prácticas DevOps en NGS:** La adopción de prácticas DevOps en NGS se plantea como un proceso gradual, orientado a estabilizar primero los elementos base de operación (roles, reglas de trabajo y trazabilidad) y luego avanzar hacia automatización y control técnico. Este enfoque evita implementaciones de golpe, reduce dependencia de prácticas individuales y permite que los equipos incorporen cambios de forma consistente entre proyectos, manteniendo la dinámica Scrum como mecanismo natural de planificación y mejora.

En la práctica, cada etapa se traduce en actividades concretas que se incorporan como trabajo técnico dentro del Sprint: se priorizan como PBIs, se planifican en Sprint Planning, se monitorean en el Daily y se ajustan en retrospectivas. El objetivo es que el equipo no ejecute un proceso paralelo, sino que integre la adopción DevOps como parte del trabajo real, dejando evidencias mínimas que permitan verificar avance y sostenibilidad.

**Tabla 59** *Plan de integración por etapas*

No	Etapas	Duración sugerida	Enfoque	Actividades clave	Responsables principales	Evidencia mínima
1	Preparación y estandarización	2-3 semanas	Alinear reglas y responsabilidades	Definir RACI por proyecto; acordar DoD técnico; publicar normas de ramas/PR/versionado; checklist mínimo de entrega	Líder técnico + SM + QA Lead + Dev Team + Gerencia TI	RACI publicado; DoD técnico; normas documentadas; repos con estructura estándar

2	Integración continua (CI) + trazabilidad	3-4 semanas	Validar cambios de forma repetible	PR obligatorio; build automático por commit/PR; validaciones automáticas visibles; trazabilidad ticket-commit-PR	DevOps Engineer + Dev Team + Líder técnico	Historial de ejecuciones CI; PR revisados; evidencia de trazabilidad completa
3	Automatización de pruebas (mínimo viable)	3-5 semanas	Reducir dependencia manual	Definir set mínimo (unit/smoke); integrarlo al pipeline; criterios mínimos por componente; fallos repetitivos - PBIs técnicos	QA/QA Lead + Dev Team + DevOps Engineer	Reportes automáticos; criterios mínimos definidos; PBIs técnicos creados
4	Entrega continua (CD) y despliegues controlados	3-5 semanas	Consistencia de releases y despliegue	Artefactos/versiones por entrega; despliegue estándar a Dev/QA; aprobación para Prod cuando aplique; rollback definido y probado; documentación mínima	DevOps Engineer + QA Lead + Líder técnico + PO	Release versionado; registros por ambiente; checklist/release note; evidencia de rollback
5	5. Monitoreo, retroalimentación y mejora continua	4 semanas	Aprender de la operación	Señales mínimas (logs/alertas/métricas); tablero básico; registro de eventos/incidentes; retro técnica - backlog de mejora	DevOps Engineer + Dev Team + QA Lead + SM	Tablero activo; bitácora de eventos/incidentes; acciones en backlog
6	Escalamiento institucional	4-6 semanas	Replicar en más proyectos	Replicar plantillas/estándares en 3-5 proyectos; sesiones de transferencia por equipo núcleo; auditorías con checklist; consolidación de guías/plantillas institucionales	Equipo núcleo + Líder técnico + PM/SM	Plantillas adoptadas; asistencia a réplicas; reportes de auditoría; guías publicadas

Nota: Elaboración propia.

Con esta ruta por etapas, NGS puede implementar las prácticas DevOps de forma controlada: primero ordena la base (roles y reglas), luego automatiza y, finalmente consolida

operación y escalamiento. El siguiente entregable aterriza las políticas operativas que normalizan la ejecución por fase del ciclo DevOps, con responsables y evidencia mínima para sostener la adopción.

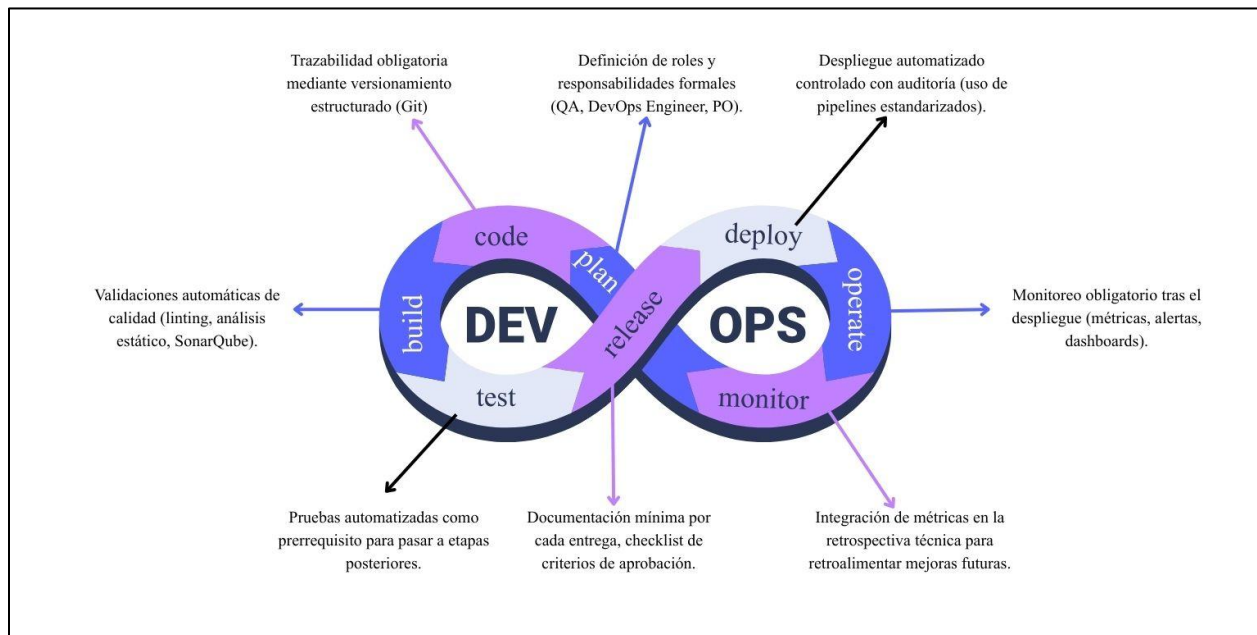
**Integración de políticas DevOps por fases del ciclo DevOps:** En el marco del modelo propuesto, se plantea un conjunto de políticas operativas mínimas asociadas al ciclo DevOps, con el propósito de asegurar consistencia entre proyectos, reducir variaciones en la forma de trabajo y fortalecer el control técnico sin afectar la dinámica ágil. Estas políticas no se presentan como controles burocráticos, sino como acuerdos institucionales que estandarizan criterios de ejecución y evidencias básicas, permitiendo que la implementación de prácticas DevOps sea sostenible y replicable dentro de NGS.

La recomendación se organiza de manera práctica: una política por etapa, enfocada en el punto donde se genera el mayor valor de control. En la fase de planificación se prioriza la claridad organizativa mediante roles definidos; durante el desarrollo se refuerza la trazabilidad con versionamiento estructurado; en construcción se incorporan validaciones automáticas para detectar errores tempranos; en pruebas se establece la automatización como requisito para avanzar; al liberar se garantiza documentación mínima y criterios de aprobación; en despliegue se busca estandarización y auditoría a través de pipelines; y finalmente, en operación se consolida el monitoreo y la retroalimentación para convertir la ejecución diaria en mejoras concretas.

Bajo esta lógica, el ciclo DevOps se convierte en un mecanismo de trabajo controlado y medible: cada etapa mantiene una intención clara, un responsable visible y una evidencia mínima asociada. Con ello, se facilita la validación interna de NGS, se mejora la trazabilidad de los cambios y se refuerza la disciplina técnica necesaria para sostener procesos CI/CD, apoyando además la integración con el marco Scrum mediante la incorporación de mejoras y hallazgos técnicos en el backlog y en retrospectivas.

En la siguiente figura se observa de manera clara y puntual la política recomendada por cada fase del ciclo DevOps:

**Figura 103** Políticas DevOps para estandarización CI/CD en el ciclo CI/CD



Nota: Elaboración propia.

La figura anterior resume las políticas mínimas propuestas para reforzar el ciclo DevOps en NGS. Sin embargo, para que estas recomendaciones sean aplicables en el día a día, es necesario traducirlas a acciones concretas, responsables visibles y evidencias verificables. Por ello, a continuación se presenta una tabla operativa que describe cómo aplicar cada política en NGS, evitando ambigüedades y facilitando su adopción por proyecto.

**Tabla 60** Tabla operativa de políticas por fase del ciclo DevOps

Fase	Política recomendada	Aplicación operativa en NGS (cómo hacerlo)	Responsable(s)	Evidencia mínima
<b>Plan</b>	Definición de roles y responsabilidades formales (QA, DevOps Engineer, PO)	Acordar responsables del pipeline, pruebas y despliegues por proyecto; dejar el RACI visible; incorporar el DoD técnico en la planificación del Sprint.	Gerencia TI / Líder técnico / SM	RACI aprobado + DoD técnico publicado
<b>Code</b>	Trazabilidad obligatoria mediante versionamiento estructurado (Git)	Establecer norma de ramas/merge; PR obligatorio; convención ticket-commit-PR; asegurar que cada cambio quede asociado a un PBI.	Dev Team / Líder técnico	Política de ramas + PR revisados + trazabilidad completa
<b>Build</b>	Validaciones automáticas de calidad (linting, análisis estático, SonarQube)	Integrar validaciones en el pipeline; ejecutar en cada cambio; exponer resultados al equipo; registrar hallazgos	DevOps Engineer / Dev Team	Reporte de validaciones por ejecución + histórico

		recurrentes como PBIs técnicos.		
<b>Test</b>	Pruebas automatizadas como prerrequisito para pasar a etapas posteriores	Definir set mínimo (unit/smoke); ejecutarlo automáticamente; acordar criterios mínimos por componente; fallos repetitivos pasan a backlog técnico.	QA Lead / Dev Team	Reporte automático de pruebas + criterios mínimos definidos
<b>Release</b>	Documentación mínima por entrega + checklist de criterios de aprobación	Preparar release note breve (qué cambia, qué validar, rollback); completar checklist de aprobación antes de liberar; mantener registro por versión.	QA Lead / PO / DevOps Engineer	Release note + checklist firmado/completado
<b>Deploy</b>	Despliegue automatizado controlado con auditoría (pipelines estandarizados)	Usar plantillas de despliegue; registrar quién/cuándo/qué versión; controlar aprobación para Prod cuando aplique; ejecutar rollback si hay falla.	DevOps Engineer / QA Lead	Registro de despliegue por ambiente + versión trazable
<b>Operate / Monitor</b>	Monitoreo obligatorio tras el despliegue (métricas, alertas, dashboards)	Definir señales mínimas; revisar post-release; registrar eventos/incidentes; convertir hallazgos en PBIs y tratarlos en retrospectiva técnica.	DevOps Engineer / Dev Team / SM	Dashboard básico + bitácora de eventos/incidentes + PBIs creados

Nota: Elaboración propia.

Una vez definida la aplicación práctica de cada política, el siguiente punto crítico es asegurar que la ejecución no dependa de interpretaciones individuales. La claridad de responsabilidades permite sostener la adopción en el tiempo, especialmente cuando participan perfiles distintos (desarrollo, QA, DevOps y roles Scrum). En este sentido, la matriz RACI se incorpora como mecanismo de gobernanza ligera para distribuir responsabilidades y definir puntos de aprobación e involucramiento.

**Tabla 61** *Matriz RACI del modelo de políticas*

Actividad clave	PO	SM	Dev Team	QA Lead / QA	DevOps Engineer	Líder técnico / Gerencia TI
Definir DoD técnico y criterios mínimos	A	C	R	R	C	C
Asegurar trazabilidad (ticket-commit-PR)	I	C	R/A	C	C	C
Configurar/administrar pipeline estándar	I	I	C	C	A/R	C

Ejecutar validaciones automáticas (build/análisis)	I	I	R	C	A/R	C
Definir y mantener pruebas automatizadas mínimas	I	I	R	A/R	C	C
Preparar release note + checklist de aprobación	A	C	C	R	C	I
Ejecutar despliegues por ambiente (automatizados)	I	I	C	C	A/R	I
Aprobar despliegue a Prod (cuando aplique)	A	I	C	C	R	A
Revisar monitoreo post-release y registrar eventos	I	C	R	R	C	I
Retrospectiva técnica y backlog de mejora	C	A/R	R	R	C	I

Nota: Elaboración propia.

En síntesis, la Gerencia de TI y el líder técnico asumen los puntos donde la matriz marca A (Aprueba y responde por el resultado), asegurando lineamientos claros, asignación de recursos y continuidad del enfoque. El equipo de desarrollo, QA/QA Lead y el Ingeniero DevOps concentran la mayor carga de ejecución marcada como R (Ejecuta la actividad), aplicando las políticas, manteniendo el pipeline y generando la evidencia técnica. De forma complementaria, cuando la matriz señala C (Es consultado y aporta criterio), estos roles participan revisando decisiones técnicas clave; y cuando aparece I (Es informado y mantiene visibilidad), se garantiza que los actores pertinentes como el Product Owner y el Scrum Master den seguimiento al avance e incorporen los ajustes necesarios al backlog.

Con las responsabilidades asignadas, se vuelve importante contar con una forma simple de verificar si las políticas se están cumpliendo de manera consistente en los proyectos. Más que un control formal, se busca un instrumento de seguimiento que permita identificar brechas y justificar decisiones, dejando registro de mejoras y acciones correctivas. Por ello, se propone un checklist de verificación que NGS puede utilizar como herramienta de validación interna.

**Tabla 62** Checklist de verificación de adopción

<b>Fase</b>	<b>Ítem de verificación</b>	<b>Cumple (Sí/Parcial/No)</b>	<b>Justificación</b>	<b>Acción correctiva</b>	<b>Responsable</b>	<b>Fecha</b>
Plan	RACI definido y disponible para el proyecto					
Plan	DoD técnico aplicado en Sprint Planning					
Code	PR obligatorio y revisado antes de integrar					
Code	Trazabilidad ticket-commit-PR en cambios relevantes					
Build	Validaciones automáticas activas en el pipeline					
Test	Pruebas automatizadas mínimas ejecutadas por entrega					
Release	Release note mínima y checklist completado por versión					
Deploy	Despliegue auditado por ambiente (registro de versión)					

Monitor	Dashboard/alertas mínimos activos y revisados post-release					
Monitor	Hallazgos convertidos en PBIs y revisados en retrospectiva					

Nota: Elaboración propia.

Con estos elementos, las políticas propuestas pasan de una recomendación conceptual a un esquema aplicable: se cuenta con una referencia visual, una guía operativa por fase, responsabilidades explícitas mediante la matriz RACI y un instrumento de verificación para dar seguimiento a su adopción. En conjunto, estos entregables facilitan que NGS pueda validar y ajustar la implementación en proyectos reales, manteniendo coherencia con el trabajo Scrum y fortaleciendo la disciplina técnica necesaria para sostener las prácticas DevOps e integrarlas con las prácticas ITIL v4 seleccionadas.

**Políticas de control de versiones para el flujo CI/CD:** Como complemento a la definición de políticas DevOps por fases del ciclo de vida, la siguiente matriz profundiza específicamente en las políticas de control de versiones y despliegue CI/CD propuestas para NGS. Este nivel de detalle permite realizar un zoom operativo sobre cómo se gestionará el código fuente, los ambientes y los despliegues, formalizando reglas claras sobre ramas, pull requests, aprobaciones, versionado y responsabilidades. De esta manera, la matriz traduce los lineamientos estratégicos definidos por fase en políticas técnicas concretas, aplicables de forma transversal en los proyectos de la organización.

El objetivo de este modelo de control de versiones y despliegue CI/CD es establecer lineamientos claros y estandarizados para la gestión del código fuente y el paso controlado de cambios entre los ambientes de desarrollo, QA, preproducción y producción en NGS. Su alcance es propositivo y referencial, orientado a servir como guía institucional para los equipos de desarrollo, sin imponer una implementación rígida, permitiendo su adaptación según el tipo de proyecto, el nivel de madurez técnica y las particularidades de cada cliente. A continuación se desarrolla la matriz con las políticas propuestas y sus beneficios:

**Tabla 63** Políticas de control de versiones y beneficios para NGS

Fase / Ambiente	Objetivo	Política de control de versiones	Tipo de integración	Responsables	Control / Aprobación	Beneficio para NGS
<b>Ramas base (institucional)</b>	Estandarizar la gestión del código	Se definen ramas institucionales: Dev, QA, Preproducción y Main (Producción).	N/A	DevOps, Líder Técnico	Reglas de rama en Azure DevOps	Consistencia y reducción de variabilidad entre proyectos
<b>Ramas de trabajo</b>	Garantizar trazabilidad por requerimiento	Cada desarrollador crea una rama de trabajo por cada User Story, Bug o Feature, basada en Main.	N/A	Desarrollador	Validación de nomenclatura y asociación a tarjeta	Trazabilidad entre requerimientos y código
<b>Desarrollo (Dev)</b>	Validar técnicamente los cambios	El desarrollador integra su rama de trabajo a Dev y despliega para pruebas unitarias.	Merge	Desarrollador	Sin aprobación formal	Detección temprana de errores
<b>Calidad (QA)</b>	Certificar funcionalidad y negocio	Los cambios maduros se integran a QA mediante Pull Request para pruebas funcionales y de aceptación.	Merge/Cherry-pick controlado	QA Lead, QA Team	Aprobación de QA	Reducción de defectos funcionales
<b>Preproducción</b>	Validar estabilidad en entorno similar a producción	Se integran únicamente los commits certificados, asociados a un pase de despliegue.	Merge/Cherry-pick controlado	DevOps, PO, Líder Técnico Asignado	Aprobación del PO o líder designado	Reducción de riesgos y conflictos entre desarrollos
<b>Producción (Main)</b>	Liberar cambios estables	Tras validar en preproducción, se despliega el release y se integra el código a Main.	Merge final	DevOps, PO	Aprobación del release y PR a Main	Estabilidad productiva y control de cambios

<b>Pull Requests (PR)</b>	Controlar la calidad del código	Todo paso a QA, Preproducción y Main debe realizarse mediante PR.	N/A	Desarrollador, Revisores	Revisión técnica y funcional	Mejora continua y colaboración
<b>Versionado de componentes</b>	Evitar colisiones y errores	Cada componente, debe tener versión única por despliegue.	N/A	DevOps	Validación automática en pipeline	Trazabilidad y soporte efectivo
<b>Correlativo de despliegue (Pase)</b>	Agrupar cambios por despliegue	Cada despliegue genera un correlativo que agrupa PR y commits incluidos.	N/A	DevOps	Registro y control del pase	Auditoría y control histórico
<b>Actualización de ramas</b>	Mantener alineación	Las ramas Dev y QA se sincronizan periódicamente con Main.	Merge	DevOps, Líder Técnico	Validación técnica	Reducción de conflictos acumulados
<b>Alineación ITIL v4</b>	Gobernar el cambio	Los despliegues y PR se gestionan como cambios controlados del servicio.	N/A	Gerencia TI, PO	Autorización formal	Gobierno y trazabilidad del servicio

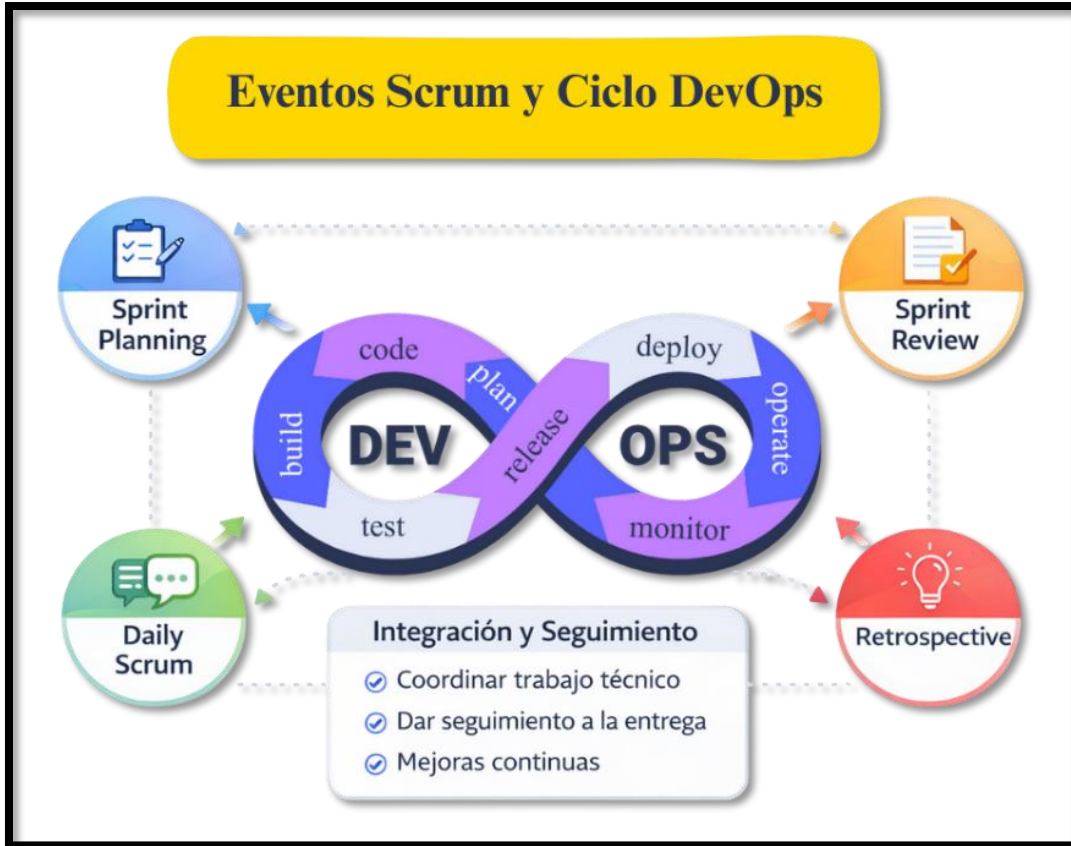
Nota: Elaboración propia.

La matriz de políticas de control de versiones consolida un marco normativo interno que busca reducir la variabilidad entre proyectos, fortalecer la trazabilidad extremo a extremo y minimizar riesgos operativos durante los despliegues. Su aplicación permite a NGS avanzar hacia un modelo CI/CD más estandarizado y gobernado, sin sacrificar la agilidad del desarrollo, alineando las prácticas técnicas con principios de control y gestión del servicio establecidos en ITIL v4. En conjunto, estas políticas constituyen un habilitador clave para la sostenibilidad, escalabilidad y mejora continua del modelo propuesto.

- **Alineación scrum con las prácticas DevOps:** Luego de definir el plan de integración de las prácticas DevOps y establecer las políticas operativas que orientan su aplicación en NGS, el siguiente paso consiste en asegurar que estas prácticas se ejecuten dentro del marco de trabajo ya adoptado por la organización. Por ello, esta sección se centra en la alineación

de las prácticas DevOps con los eventos de Scrum, utilizando Sprint Planning, Daily Scrum, Sprint Review y Retrospective como puntos naturales para coordinar el trabajo técnico, dar seguimiento a la entrega y convertir los hallazgos en mejoras continuas.

**Figura 104** Integración de los eventos scrum y el ciclo DevOps



Nota: Elaboración propia.

A continuación se presenta la tabla de alineación entre los eventos de Scrum y las prácticas DevOps propuestas para NGS. El objetivo de esta matriz es mostrar, de forma concreta, en qué momento del ciclo Scrum se activa cada práctica técnica (integración, pruebas, entrega, monitoreo y colaboración) y qué resultados se espera obtener en cada punto de control. Con ello, se evita tratar DevOps como un flujo paralelo y se refuerza la idea de que la adopción se gestiona dentro del Sprint, utilizando los eventos de Scrum como espacios naturales para coordinar, verificar avances y convertir hallazgos en mejoras continuas.

**Tabla 64** Alineación del modelo Scrum con las prácticas DevOps en el ciclo del Sprint

Evento Scrum	Prácticas DevOps integradas (modelo propuesto)	¿Cómo se aplica en el modelo?	Resultado esperado
<b>Sprint Planning</b>	Gestión colaborativa del flujo, Control de versiones (VCS), Integración continua (CI), Automatización de pruebas (criterios mínimos)	En la planificación del Sprint se definen criterios técnicos mínimos para completar PBIs (trazabilidad, validaciones automáticas y pruebas mínimas). Se incluyen PBIs técnicos cuando se requiera ajustar pipeline, versionado o pruebas.	Sprint Backlog con PBIs trazables + criterios técnicos mínimos establecidos (DoD técnico).
<b>Daily Scrum</b>	Integración continua (CI), Automatización de pruebas, Gestión colaborativa del flujo	El seguimiento diario incorpora el estado del CI como insumo de control: fallos de build, conflictos de integración y pruebas fallidas se tratan como impedimentos y generan acciones inmediatas.	Impedimentos técnicos atendidos temprano + continuidad del flujo durante el Sprint.
<b>Sprint Review</b>	Entrega continua (CD), Automatización de pruebas, Monitoreo y retroalimentación	La revisión valida el incremento con evidencia básica: versión identificable cuando aplique, resultados de pruebas y retroalimentación registrada. Los ajustes se transforman en PBIs funcionales o técnicos.	Incremento validado + feedback registrado + PBIs incorporados al Product Backlog.
<b>Sprint Retrospective</b>	Monitoreo y retroalimentación, Gestión colaborativa del flujo, Mejora técnica de CI/CD y pruebas	La retrospectiva incorpora revisión de fallas recurrentes y trabajo manual; los hallazgos se convierten en PBIs técnicos priorizables para mejorar pipeline, pruebas y estándares.	Acciones de mejora acordadas + PBIs técnicos creados y priorizados.

Nota: Elaboración propia.

En conjunto, la alineación por eventos Scrum evidencia que las prácticas DevOps del modelo se incorporan en puntos naturales de control del Sprint: se definen criterios técnicos en la planificación, se gestionan impedimentos a diario, se valida el incremento en la revisión y se convierten hallazgos en mejoras durante la retrospectiva. Esto permite que la adopción DevOps no opere como un esfuerzo paralelo, sino como parte del ciclo de inspección y adaptación propio de

Scrum.

A partir de esa lógica del “cuándo ocurre”, resulta necesario precisar también el componente organizativo “quién lo sostiene”. Por ello, la siguiente tabla relaciona los roles Scrum con las prácticas DevOps, aclarando el tipo de aporte que cada rol realiza para mantener el flujo, respaldar decisiones y asegurar continuidad en la implementación del modelo.

**Tabla 65** Alineación de roles Scrum con las prácticas DevOps del modelo propuesto

Rol Scrum	En qué prácticas DevOps influye principalmente	Aporte clave dentro del modelo	Resultado esperado
<b>Product Owner (PO)</b>	Entrega continua; Monitoreo y retroalimentación	Prioriza en el Product Backlog las mejoras técnicas necesarias y valida incrementos considerando feedback y estabilidad	Backlog equilibrado (valor + sostenibilidad) y decisiones de priorización claras
<b>Scrum Master (SM)</b>	Gestión colaborativa del flujo; Integración continua	Facilita acuerdos de trabajo, visibiliza impedimentos que afectan el flujo y promueve mejora continua Sprint a Sprint	Menos bloqueos recurrentes y mayor consistencia del proceso
<b>Equipo de desarrollo (Dev Team)</b>	Control de versiones; Integración continua; Automatización de pruebas	Ejecuta la disciplina técnica requerida (integración frecuente, validación y calidad), asegurando evidencia mínima en cada incremento	Incrementos verificables con menor dependencia manual y menos retrabajo

Nota: Elaboración propia.

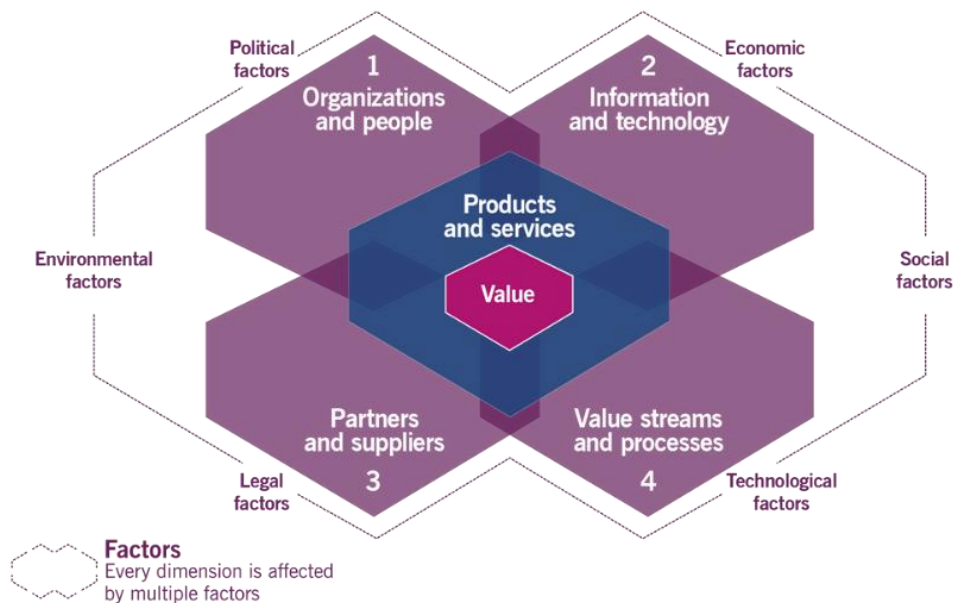
La integración propuesta requiere que cada rol Scrum contribuya desde su función natural: el Product Owner asegura prioridad y trazabilidad de las mejoras técnicas dentro del backlog, el Scrum Master protege el flujo de trabajo y facilita la eliminación de impedimentos, y el equipo de desarrollo ejecuta la disciplina técnica que hace posible la integración, validación y entrega continua. Esta claridad de aportes complementa la alineación por eventos del Sprint y refuerza la sostenibilidad del modelo en NGS.

Con la alineación entre Scrum y DevOps definida desde los roles y los eventos del Sprint,

el modelo queda integrado al marco ágil que NGS utiliza para planificar y ejecutar el trabajo. No obstante, para asegurar que esta integración se mantenga sostenible y consistente entre proyectos, es necesario complementarla con la perspectiva de gestión de servicios. En ese sentido, ITIL v4 aporta un marco orientado a valor y control práctico, permitiendo fortalecer aspectos como trazabilidad de cambios, consistencia en despliegues, monitoreo y mejora continua sin desplazar la agilidad de Scrum.

- **Alineación con ITIL V4:** En esta etapa se incorpora ITIL v4 como elemento de soporte para la implementación de prácticas DevOps en NGS. El propósito es utilizar componentes del Sistema de Valor del Servicio, principios guía, prácticas identificadas y el modelo de cuatro dimensiones para asegurar que la adopción DevOps considere personas, tecnología, proveedores y procesos. De esta manera, el modelo propuesto consolida una integración completa: Scrum organiza el trabajo, DevOps operacionaliza la entrega continua e ITIL v4 aporta criterios de gestión para sostener la mejora y la estabilidad del servicio.

**Figura 105** Modelo de las cuatro dimensiones de la gestión del servicio de ITIL 4



Nota: Adaptado de ITIL Foundation: ITIL 4 Edition, por AXELOS Limited, 2019.

Como punto de partida para alinear la propuesta CI/CD con ITIL v4, se utilizarán dos componentes clave del marco: el Modelo de las Cuatro Dimensiones de la Gestión del Servicio y

los Principios Guía. Las cuatro dimensiones permiten estructurar la implementación considerando integralmente organizaciones y personas, información y tecnología, socios y proveedores, y flujos de valor y procesos; mientras que los principios guía funcionan como criterios prácticos para tomar decisiones durante la adopción, priorizando valor, simplicidad, colaboración, retroalimentación y automatización. Con ello, la integración DevOps se plantea como una mejora sostenible y consistente, sin limitarse únicamente a herramientas o actividades técnicas.

La matriz que se muestra a continuación se estructuró tomando como base los hallazgos/brechas priorizadas identificadas en el diagnóstico (capítulo 4) y, para cada una, se seleccionaron los principios guía de ITIL v4 que orientan la toma de decisiones. Luego, esas brechas se tradujeron en acciones concretas organizadas según las cuatro dimensiones (organizaciones y personas; información y tecnología; socios y proveedores; y flujos de valor y procesos), de manera que cada mejora propuesta no quede limitada a lo técnico, sino que contemple roles, capacidades, dependencias y ajustes de proceso necesarios para su adopción en NGS.

**Tabla 66** Principios guía más 4 dimensiones aplicada a la integración DevOps en NGS

Escenario en NGS (contexto del modelo)	Principios guía ITIL v4 aplicables	Organizaciones y personas	Información y tecnología	Socios y proveedores	Flujos de valor y procesos
<b>E1. Variabilidad del flujo CI/CD entre proyectos y dependencia de ejecución manual</b>	Enfocarse en el valor; Empezar donde estás; Mantenerlo simple y práctico	Definir roles base (Dev, QA, DevOps) y responsables por proyecto; acuerdos de trabajo (DoD técnico)	Estandarizar pipeline base; versionado estructurado; pruebas mínimas integradas	Asegurar accesos y soporte a herramientas (repositorios, CI/CD, análisis)	Definir flujo mínimo CI/CD y checklist de adopción; revisión periódica de cumplimiento
<b>E2. Falta de trazabilidad consistente (cambio-código-entrega)</b>	Colaborar y promover visibilidad; Pensar y trabajar holísticamente	Capacitación en buenas prácticas Git/PR; disciplina de revisión de código	Regla ticket-commit-PR; registros de builds y despliegues	Alineación con políticas del cliente/proveedor (si aplica)	Integrar trazabilidad al proceso Scrum (PBI - PR - release) y reportarla en Review

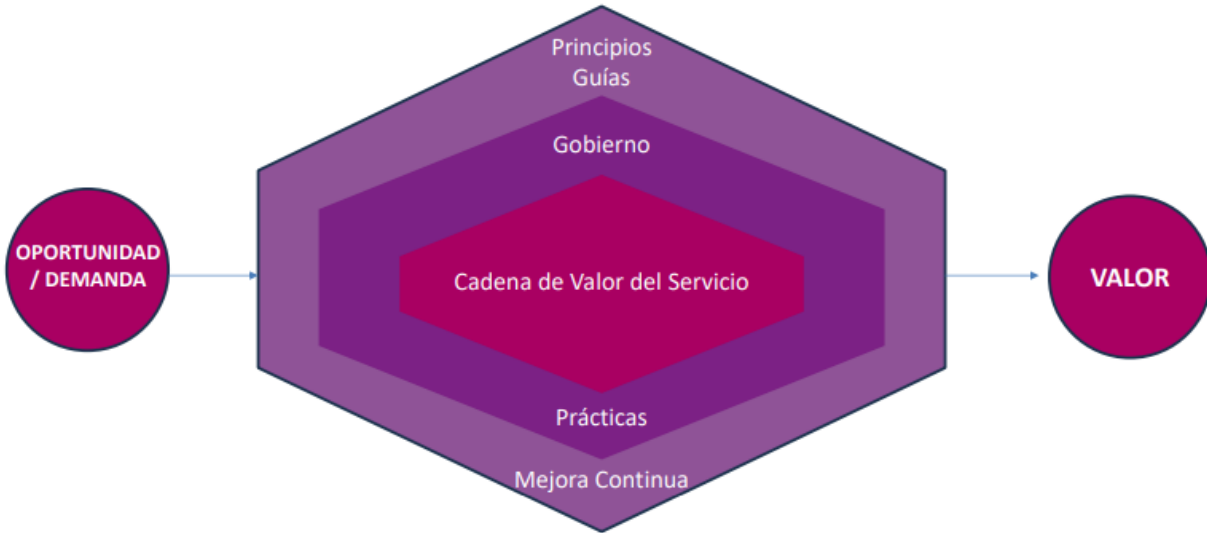
<b>E3. Calidad inconsistente por ausencia o bajo nivel de pruebas automatizadas</b>	Progresar iterativamente con retroalimentación; Optimizar y automatizar	Definir estándar mínimo de pruebas; liderazgo de QA/QA Lead y acompañamiento al equipo	Integrar ejecución automática de pruebas (unit/smoke) al pipeline; reportes visibles	Selección de herramientas/librerías de pruebas según stack	Incluir criterios de pruebas en DoD; fallos recurrentes se convierten en PBIs técnicos
<b>E4. Despliegues con control limitado y recuperación poco clara</b>	Mantenerlo simple y práctico; Optimizar y automatizar; Enfocarse en el valor	Definir responsables de despliegue y validación por ambiente	Pipelines estandarizados para despliegue; registro de versión por ambiente	Coordinación con infraestructura/plataformas (on-prem/cloud)	Establecer pasos de liberación y aprobación cuando aplique; definición clara de rollback
<b>E5. Monitoreo post-entrega débil y retroalimentación poco estructurada</b>	Progresar iterativamente con retroalimentación; Colaborar y promover visibilidad	Disciplina de revisión post-release; retro técnica en Retrospective	Dashboard básico + alertas/logs mínimos; bitácora de eventos/incidentes	Integración con herramientas de monitoreo existentes o del cliente	Feedback se convierte en PBIs; revisión de incidentes/problemas y acciones de mejora

Nota: Elaboración propia.

Esta matriz permite pasar del diagnóstico a la acción, conectando cada brecha priorizada con criterios de decisión (principios guía) y con medidas concretas organizadas por las cuatro dimensiones de ITIL v4. Con ello, la integración de prácticas DevOps no se limita a ajustes técnicos, sino que incorpora responsabilidades, capacidades, dependencias y procesos necesarios para sostenerla en NGS. A partir de esta base integral, el siguiente paso es representar la implementación como un flujo de valor mediante la Cadena de Valor del Servicio, mostrando cómo se articulan Scrum, DevOps e ITIL v4 en una secuencia operativa de entrega y mejora continua.

Luego de estructurar la propuesta con base en las cuatro dimensiones y los principios guía de ITIL v4, el siguiente paso consiste en aplicar el Sistema de Valor del Servicio (SVS) como marco de alineación. Esto permite conectar la ejecución técnica de DevOps con la lógica de gestión de servicios, asegurando que la entrega continua no dependa únicamente de herramientas, sino que se soporte en prácticas de ITIL v4 relacionadas con cambios, despliegues, monitoreo, incidentes, configuración y mejora continua. En consecuencia, a continuación se presenta el flujo de valor (demanda-valor) utilizando la Cadena de Valor del Servicio, incorporando en cada actividad la práctica DevOps correspondiente y la práctica ITIL v4 que respalda su control.

**Figura 106** El Sistema de valor del servicio en ITIL 4



Fuente: Axelos, "Fundamentos de ITIL: Edición ITIL4" (2019) - Sistema de valor del servicio de ITIL 4.

Este flujo de valor se presenta con el formato operativo de la Cadena de Valor del Servicio, iniciando en la demanda y finalizando en la generación de valor, tal como se utiliza en ejercicios prácticos de ITIL v4. La estructura permite describir, paso a paso, cómo se articula el trabajo gestionado con Scrum con las prácticas DevOps definidas para NGS, incorporando además un punto de seguimiento posterior al valor para asegurar continuidad y mejora.

**Nombre del Flujo de Valor:** Entrega y despliegue controlado de software mediante CI/CD (NGS)

**Tabla 67** Flujo de Valor: Entrega y despliegue controlado de software mediante CI/CD (NGS)

Entrada / Salida	Actividad Clave Cadena de Valor del Servicio (SVC)	Práctica	Rol	Actividad
DEMAND	-	-	-	Solicitud de cambio / necesidad de entrega registrada como PBI o requerimiento técnico priorizable.

-	<b>Engage</b>	<b>Gestión colaborativa del flujo</b>	PO, SM, Dev Team, QA Lead	Alineación inicial: alcance, criterios de aceptación, dependencias y responsables de validación. Se asegura visibilidad del trabajo y riesgos.
-	<b>Plan, Engage</b>	<b>Gestión colaborativa del flujo</b>	PO, SM, Líder técnico	Planificación del Sprint incorporando PBIs técnicos necesarios (pipeline, pruebas, monitoreo). Se definen criterios técnicos mínimos del incremento.
-	<b>Design &amp; Transition</b>	<b>Control de versiones (VCS)</b>	Líder técnico, Dev Team	Definición de estrategia de ramas/PR y trazabilidad PBI-commit-PR. Se establecen reglas mínimas para integrar cambios.
-	<b>Obtain/Build</b>	<b>Integración continua (CI)</b>	Dev Team, Ingeniero DevOps	Construcción automatizada y validaciones básicas en cada integración. Se detectan fallos temprano para evitar acumulación de defectos.
-	<b>Obtain/Build</b>	<b>Automatización de pruebas</b>	QA Lead, Dev Team	Ejecución de pruebas automatizadas mínimas (unitarias/smoke según aplique). Fallos se corrigen o se registran como PBIs técnicos.
-	<b>Design &amp; Transition, Obtain/Build</b>	<b>Infraestructura como código (IaC)</b>	Ingeniero DevOps, Apoyo Infra (si aplica)	Preparación de ambientes consistentes mediante scripts/plantillas; reducción de errores por configuración manual.

-	<b>Deliver &amp; Support</b>	<b>Entrega continua (CD)</b>	Ingeniero DevOps, QA Lead	Despliegue por ambientes (Dev/QA/Prod si aplica) con versión identificable y control del pase. Validación post-despliegue y registro del resultado.
-	<b>Deliver &amp; Support</b>	<b>Monitoreo y retroalimentación</b>	DevOps, Dev Team, QA Lead	Observación post-release: métricas/alertas/logs básicos. Se registra retroalimentación técnica y funcional para el backlog.
-	<b>Improve</b>	<b>Gestión colaborativa del flujo</b>	SM, Dev/QA/DevOps, PO	Retro técnica: análisis de fricciones del flujo (manualidad, fallas repetidas) y priorización de PBIs de mejora para el siguiente Sprint.
<b>VALUE</b>	-	-	-	Incremento entregado y estable: versión trazable, validación concluida y feedback integrado al backlog.
-	<b>Engage (Seguimiento)</b>	<b>Monitoreo y retroalimentación</b>	PO, SM, Líder técnico, QA Lead	Revisión periódica de adopción: resultados de entrega, fallas recurrentes, mejoras aplicadas y ajustes de políticas/prácticas para sostener el modelo.

Nota: Elaboración propia.

En conjunto, el flujo de valor organiza la integración CI/CD como una secuencia completa y comprensible, evitando que DevOps se perciba como actividades sueltas o dependientes de esfuerzos individuales. Al incorporar un punto de seguimiento posterior, se refuerza la sostenibilidad del modelo, asegurando que los aprendizajes de cada entrega se transformen en ajustes concretos al Product Backlog y al propio flujo de trabajo gestionado con Scrum.

- Plan de integración de prácticas DevOps alineado a prácticas ITIL v4:** Luego de definir el flujo de valor mediante el SVS y su Cadena de Valor del Servicio, se presenta un plan de integración que vincula de forma directa las prácticas DevOps identificadas para NGS con las prácticas clave de ITIL v4 aplicables a la gestión de servicios. El objetivo de esta sección es traducir la alineación conceptual en acciones operativas concretas, de manera que cada práctica DevOps tenga un respaldo de gestión (cambios, despliegues, monitoreo, incidentes, configuración, problemas y mejora continua) sin agregar complejidad innecesaria.

El plan se estructura como una matriz de correspondencia práctica-práctica: para cada práctica DevOps se identifican las prácticas ITIL v4 que mejor la soportan y se describe cómo se aplicaría en NGS mediante acciones observables, responsables y evidencias cualitativas. Con ello, la integración deja de depender de interpretaciones individuales y se convierte en una guía consistente para implementar, sostener y mejorar el modelo de entrega continua.

**Tabla 68** *Plan de integración DevOps ITIL v4*

Práctica DevOps (NGS)	Práctica(s) ITIL v4 relacionada	Aplicación operativa en NGS	Responsables	Evidencia cualitativa
Integración continua (CI)	Monitoreo y gestión de eventos; Gestión de cambios	Ejecutar integración frecuente con validaciones automatizadas en el pipeline; registrar fallos relevantes y tratarlos como impedimentos técnicos del Sprint	Dev Team + DevOps + SM	Historial de ejecuciones del pipeline + registro de fallos/resolución
Entrega continua (CD)	Gestión de despliegues; Gestión de cambios	Promover versiones por ambientes con control del despliegue y pasos definidos; asegurar trazabilidad de la liberación y validación posterior	DevOps + QA Lead	Bitácora de despliegues + versión identificable por ambiente

Control de versiones (VCS)	Gestión de configuración y activos; Gestión de cambios	Estandarizar ramas, PR y trazabilidad PBI-commit-PR; asegurar revisión antes de integrar a ramas principales	Líder técnico + Dev Team	Guía aplicada + PR con referencia a PBIs y evidencia de revisión
Infraestructura como código (IaC)	Gestión de configuración y activos; Gestión de despliegues	Reducir configuración manual usando scripts/plantillas para entornos Dev/QA; mantener versiones de configuración bajo control	DevOps + Infra (si aplica)	Repositorio con plantillas/scripts + consistencia de entornos
Automatización de pruebas	Gestión de problemas; Mejora continua	Incorporar pruebas mínimas automatizadas en el pipeline; fallas repetitivas se convierten en PBIs técnicos de mejora	QA Lead + Dev Team	Reportes de pruebas + PBIs técnicos derivados de fallas
Monitoreo y retroalimentación	Monitoreo y gestión de eventos; Gestión de incidentes; Mejora continua	Monitoreo básico post-release (logs/alertas/métricas); convertir hallazgos en acciones de mejora del backlog	DevOps + Equipo	Bitácora de eventos/incidencias + acciones de mejora registradas
Gestión colaborativa del flujo	Mejora continua; Gestión de cambios	Asegurar visibilidad y coordinación del flujo en eventos Scrum; mantener acuerdos operativos mínimos y ajustar el proceso con retrospectivas	PO + SM + Equipo	Minutas de retro técnica + acuerdos publicados y aplicados

Nota: Elaboración propia.

Este plan de integración establece un puente operativo entre la ejecución técnica de DevOps y el enfoque de gestión de servicios de ITIL v4. La matriz clarifica qué prácticas ITIL respaldan cada práctica DevOps y cómo se aplican en el contexto de NGS mediante responsables y evidencias observables. Con esta base, el modelo no solo describe el “qué” de la integración, sino que orienta el “cómo” para sostener consistencia, control y mejora continua; a partir de aquí, el cronograma de implementación permite organizar su adopción gradual en proyectos piloto y su

expansión progresiva.

- **Lineamientos de arquitectura que habilitan el modelo CI/CD (seguridad, escalabilidad y trazabilidad):** La integración entre Scrum, prácticas DevOps e ITIL v4, requiere condiciones de arquitectura que sostengan su operación de forma consistente. En el diagnóstico se identificaron retos vinculados a la falta de uniformidad entre proyectos, controles de seguridad no integrados al flujo y trazabilidad incompleta entre cambios, compilaciones y despliegues, lo cual limita la auditabilidad y eleva el riesgo operativo. En esta sección se establecen lineamientos prácticos para que la arquitectura soporte el ciclo CI/CD sin introducir complejidad innecesaria.

**Tabla 69** Matriz de lineamientos arquitectónicos para soportar CI/CD en NGS

Eje arquitectónico	Situación observada	Lineamiento propuesto	Responsables	Evidencia observable
<b>Trazabilidad end-to-end</b>	Trazabilidad fragmentada entre cambios, builds y despliegues	Definir trazabilidad mínima obligatoria: PBI/ticket - PR - build - release (registro de versión y ambiente). Asegurar que cada despliegue quede asociado a un artefacto versionado.	Líder técnico + DevOps + QA Lead	PR con referencia + ejecución de pipeline vinculada + bitácora de release por ambiente
<b>Seguridad integrada al flujo</b>	Seguridad no integrada al pipeline	Incorporar controles mínimos en el flujo: revisión obligatoria, manejo de secretos y validaciones técnicas según política, antes de promover a ambientes críticos.	DevOps + Líder técnico + QA Lead	Evidencia de revisión/aprobación + reglas de repositorio + pasos definidos en pipeline/release
<b>Estandarización entre entornos</b>	Diferencias entre entornos y procesos manuales en despliegues	Mantener coherencia Dev/QA/Preprod/Prod mediante plantillas y configuración controlada (scripts/lineamientos). Evitar configuraciones “por proyecto” sin justificación.	DevOps + QA + Infra (si aplica)	Guías/plantillas publicadas + despliegues repetibles + registro de excepciones

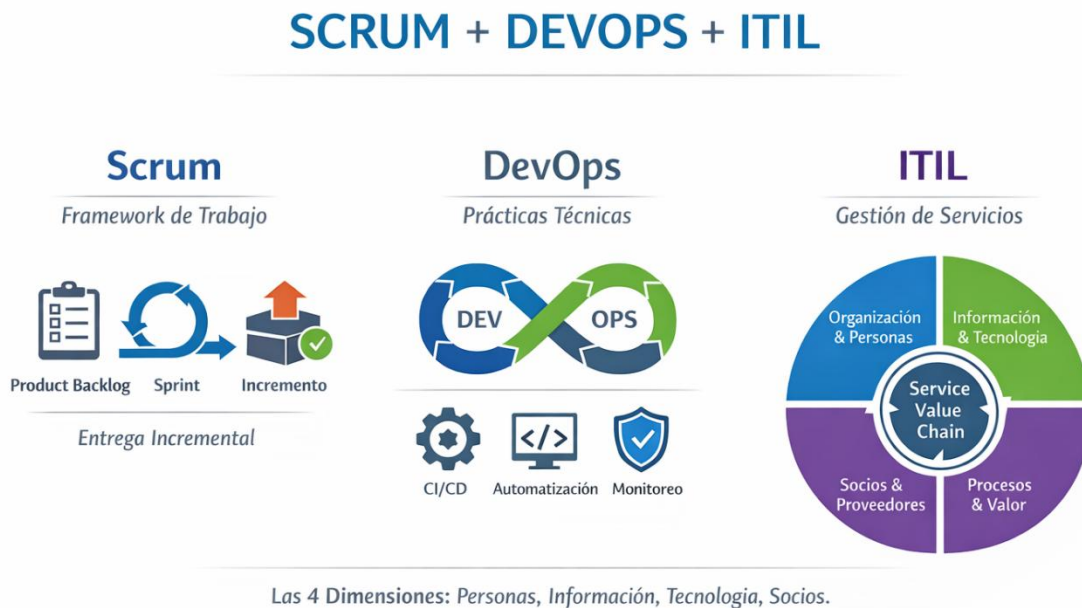
<b>Escalabilidad operativa</b>	Arquitectura con poca elasticidad / limitaciones para escalar	Implementación gradual: iniciar con pilotos, consolidar plantillas y luego extender; priorizar automatización que reduzca retrabajo y tiempos de entrega antes de cambios mayores de infraestructura.	Gerencia TI + PM + DevOps	Plan por fases + adopción en pilotos + lecciones aprendidas y ajustes
<b>Observabilidad post-release</b>	Monitoreo irregular según proyecto/cliente	Definir monitoreo mínimo post-despliegue (logs/alertas/métricas básicas) y retroalimentación al backlog para sostener mejora continua.	DevOps + Equipo	Bitácora de eventos + alertas activas + PBIs técnicos derivados

Nota: Elaboración propia.

Estos lineamientos permiten que el modelo no quede únicamente a nivel de prácticas, sino que esté respaldado por condiciones mínimas de arquitectura centradas en trazabilidad, seguridad, estandarización y operación. Con ello, la integración CI/CD se vuelve más consistente entre proyectos y más fácil de sostener en el tiempo. A partir de esta base, el cronograma organiza la implementación gradual, comenzando con pilotos y consolidando la adopción de forma progresiva.

La siguiente figura resume la lógica del modelo propuesto, Scrum organiza el trabajo mediante backlog y eventos del Sprint; DevOps aporta las prácticas que habilitan automatización, CI/CD y retroalimentación técnica; e ITIL v4 enmarca la adopción desde la gestión del servicio, integrando las cuatro dimensiones y el flujo de valor para sostener consistencia y mejora continua en NGS.

Figura 107 Resumen del plan de integración de Scrum - DevOps - ITIL V4



Nota: Elaboración propia.

- **Plan de gestión del cambio organizacional para la adopción de las metodologías:** La adopción del modelo de integración de prácticas DevOps, metodologías ágiles y lineamientos de ITIL v4 en NGS implica transformaciones que van más allá del componente tecnológico, abarcando cambios en la forma de trabajo, en los roles organizacionales y en la cultura de la organización. Estos cambios requieren ser gestionados de manera estructurada para asegurar una transición ordenada, minimizar la resistencia al cambio y garantizar la sostenibilidad del modelo propuesto.

En este contexto, se desarrolla el presente entregable de gestión del cambio organizacional, concebido como un componente habilitador del modelo DevOps-Ágil-ITIL v4. Su finalidad es acompañar a la organización durante el proceso de adopción del modelo, facilitando la alineación del personal, la preparación de los equipos y la incorporación progresiva de las nuevas prácticas de trabajo.

**Objetivo del entregable:**

Establecer un marco estructurado de gestión del cambio organizacional que permita a NGS adoptar de manera progresiva y sostenible el modelo de integración de prácticas DevOps,

metodologías ágiles y lineamientos de ITIL v4, asegurando la participación del personal, reduciendo la resistencia al cambio y fortaleciendo la cultura de mejora continua.

**Alcance del entregable:**

El presente entregable aplica a las áreas y equipos involucrados en el desarrollo, despliegue y gestión de servicios de TI en NGS, incluyendo equipos de desarrollo, aseguramiento de calidad, operaciones, liderazgo técnico y gestión de proyectos.

La gestión del cambio organizacional propuesta se estructura en cinco fases secuenciales y complementarias, orientadas a facilitar la adopción progresiva del modelo DevOps-Ágil-ITIL v4 en NGS. Estas fases comprenden la sensibilización y alineación organizacional, enfocada en generar conciencia sobre la necesidad del cambio; la preparación organizacional, orientada a definir roles, responsabilidades y equipos piloto; la capacitación y acompañamiento, destinada a fortalecer las competencias técnicas y metodológicas; la implementación gradual del cambio, que permite aplicar el modelo de forma controlada en proyectos piloto; y finalmente, el seguimiento y mejora continua, cuyo objetivo es asegurar la sostenibilidad del cambio mediante la revisión periódica de resultados y ajustes al modelo propuesto.

**Tabla 70** *Matriz de las fases de la gestión del cambio organizacional*

Fase del cambio	Objetivo	Actividades clave	Roles responsables	Entregables	Indicadores de adopción
Sensibilización y alineación	Generar conciencia organizacional sobre la necesidad e impacto del cambio	<ul style="list-style-type: none"> <li>Socialización del modelo propuesto</li> <li>Comunicación de beneficios y riesgos asociados</li> </ul>	Dirección / Gerencia Equipo Núcleo de Cambio	<ul style="list-style-type: none"> <li>Material informativo</li> <li>Registro de sesiones de socialización</li> </ul>	<ul style="list-style-type: none"> <li>Nivel de participación del personal</li> <li>Retroalimentación inicial</li> </ul>
Preparación organizacional	Preparar a los equipos para la adopción del cambio	<ul style="list-style-type: none"> <li>Definición de nuevos roles y responsabilidades</li> <li>Selección de equipos piloto</li> </ul>	Dirección / Gerencia Líder DevOps Scrum Master	<ul style="list-style-type: none"> <li>Matriz de roles y responsabilidades</li> <li>Lista de equipos piloto</li> </ul>	<ul style="list-style-type: none"> <li>Roles formalizados</li> <li>Equipos piloto definidos</li> </ul>
Capacitación y acompañamiento	Desarrollar competencias técnicas y metodológicas	<ul style="list-style-type: none"> <li>Talleres prácticos de DevOps y Scrum</li> <li>Acompañamiento durante los primeros sprints</li> </ul>	Líder DevOps Scrum Master / Facilitador Ágil	<ul style="list-style-type: none"> <li>Plan de capacitación</li> <li>Materiales formativos</li> </ul>	<ul style="list-style-type: none"> <li>Participación en capacitación</li> <li>Nivel de comprensión del modelo</li> </ul>

Implementación gradual del cambio	Aplicar el modelo de forma controlada y progresiva	<ul style="list-style-type: none"> <li>• Uso del pipeline CI/CD en proyectos piloto</li> <li>• Aplicación de checklists de adopción</li> </ul>	Equipos técnicos Líder DevOps	<ul style="list-style-type: none"> <li>• Evidencia de uso de CI/CD</li> <li>• Checklists de adopción completados</li> </ul>	<ul style="list-style-type: none"> <li>• Proyectos que utilizan CI/CD</li> <li>• Frecuencia de despliegues</li> </ul>
Seguimiento y mejora continua	Asegurar la sostenibilidad del cambio	<ul style="list-style-type: none"> <li>• Revisión periódica de indicadores</li> <li>• Retrospectivas organizacionales</li> </ul>	Dirección / Gerencia Scrum Master	<ul style="list-style-type: none"> <li>• Reportes de indicadores</li> <li>• Planes de mejora continua</li> </ul>	<ul style="list-style-type: none"> <li>• Cumplimiento de indicadores</li> <li>• Nivel de madurez organizacional</li> </ul>

Nota: Elaboración propia.

La matriz integral de gestión del cambio organizacional presentada constituye un entregable clave para la aplicabilidad del modelo propuesto, al permitir gestionar de forma ordenada los aspectos humanos, organizacionales y culturales asociados a la adopción de nuevas metodologías de trabajo en NGS. Este enfoque contribuye a reducir los riesgos de implementación, facilita la apropiación del modelo por parte del personal y refuerza la sostenibilidad de la propuesta en el tiempo.

## 6.5 MEDIDAS DE CONTROL

Los controles se estructuran en un conjunto de métricas operativas del pipeline (KPIs DORA), métricas de calidad y seguridad, métricas de trazabilidad y gobernanza ligera, e indicadores de adopción y satisfacción. Cada indicador incluye su frecuencia de medición, herramienta de recolección y límites aceptables, con el fin de asegurar seguimiento periódico y toma de decisiones basada en datos.

### 6.5.1 PLAN DE INTEGRACIÓN DE MÉTRICAS DORA COMO PARTE DEL PIPELINE DE ENTREGA CONTINUA PARA UNA GESTIÓN BASADA EN DATOS

El plan de integración de métricas DORA como parte del pipeline de entrega continua constituye un instrumento clave para operacionalizar la medición objetiva del desempeño del flujo CI/CD en la consultora NGS. En el contexto de esta propuesta, dicha matriz permite vincular de forma estructurada los indicadores DORA con las prácticas DevOps, las actividades del pipeline y los objetivos de entrega continua definidos, transformando los resultados del diagnóstico en mecanismos concretos de control y seguimiento.

**Tabla 71** Implementación por fases de las métricas DORA

Fase	Actividad Principal	Responsable	Resultado
Fase 1	Definición formal de métricas DORA	Líder DevOps	Métricas estandarizadas
Fase 2	Integración de fuentes de datos CI/CD	Equipo DevOps	Recolección automática
Fase 3	Diseño de tableros de control	DevOps / QA	Visualización en tiempo real
Fase 4	Capacitación del personal	Gestión TI	Uso efectivo de KPIs
Fase 5	Revisión periódica de métricas	Dirección TI	Mejora continua basada en datos

Nota: Elaboración propia.

La integración de métricas DORA facilita evaluar el impacto real de la automatización, la calidad del software y la estabilidad del servicio, asegurando coherencia entre la ejecución operativa y los objetivos estratégicos de la organización. En respuesta a esta brecha, el presente plan operacionaliza la recomendación del Capítulo V de institucionalizar un sistema de KPIs orientados a la entrega continua, adoptando métricas DORA como estándar de referencia, integrando las métricas DORA directamente al pipeline de entrega continua, con el fin de fortalecer la toma de decisiones basada en datos, mejorar la transparencia del servicio y reducir la dependencia de percepciones subjetivas identificadas en los resultados de la investigación.

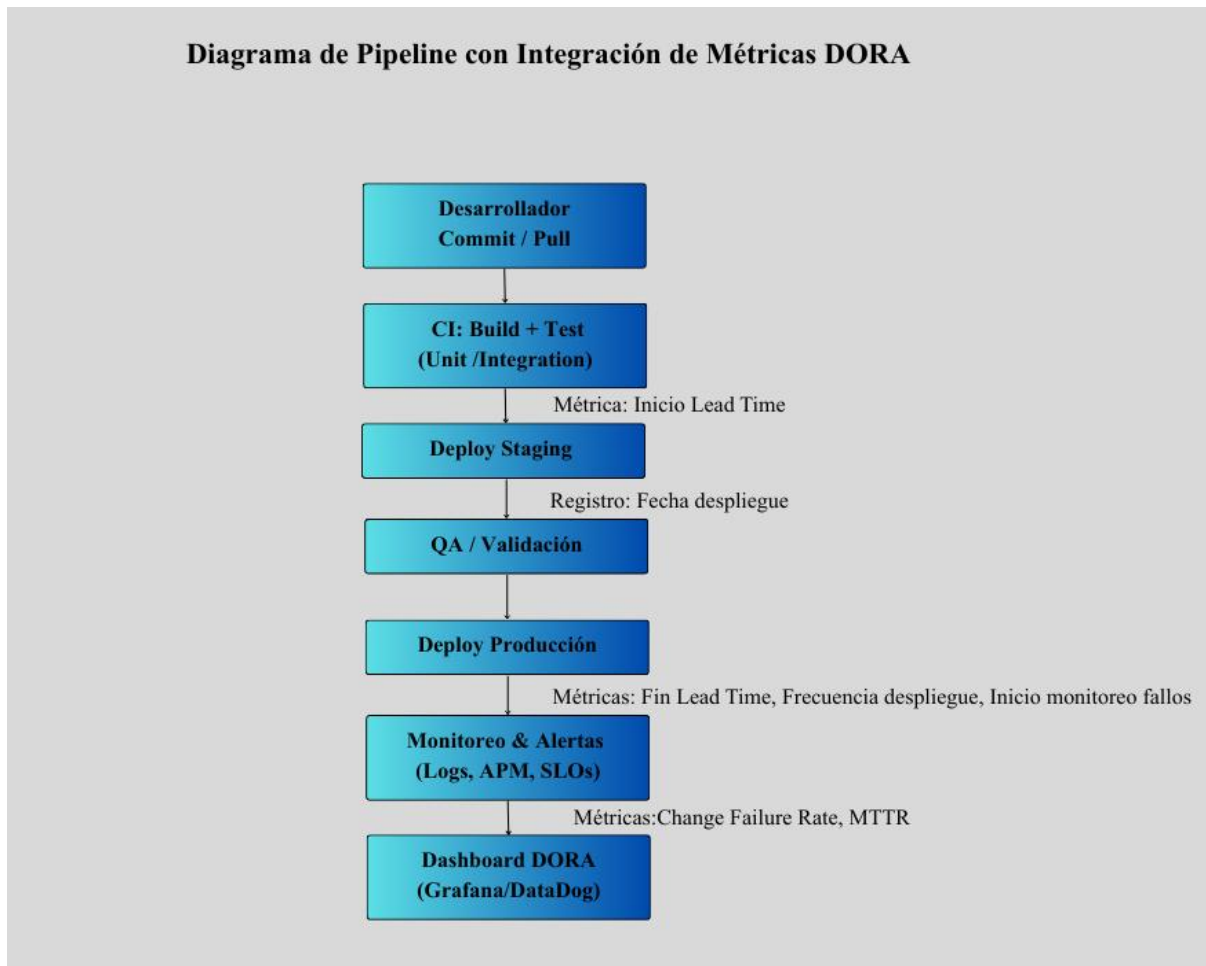
El alcance del plan contempla:

1. La definición de métricas DORA aplicables al contexto de NGS.
2. La integración técnica de dichas métricas al pipeline CI/CD.
3. El diseño de tableros automatizados de seguimiento.
4. La incorporación de las métricas en la gestión ágil, DevOps e ITIL V4.
5. La capacitación y uso organizacional de los indicadores para la toma de decisiones.

El pipeline con integración de métricas DORA sirve para tener una visión clara y completa de todo el proceso de desarrollo, desde que un desarrollador hace un commit hasta que el cambio llega a producción y se monitorea su impacto. Al incorporar métricas como el *lead time*, la *frecuencia de despliegue*, la *tasa de fallos por cambio* y el *MTTR*, podemos medir de forma

objetiva la eficiencia y la calidad del flujo de trabajo, detectar cuellos de botella, aumentar la confianza en los despliegues y fomentar la mejora continua. Se convierte el ciclo de entrega de software en un sistema transparente y basado en datos, que ayuda al equipo de NGS a entregar valor más rápido, con menos riesgos y alineado con las mejores prácticas de DevOps.

**Figura 108** Diagrama de pipeline con integración de métricas DORA entre las diferentes fases



Nota: Elaboración propia.

1. **Desarrollador: commit y pull request:** Aquí el desarrollador sube sus cambios al repositorio y abre un PR para revisión. Es el punto de partida del *lead time*, porque desde este momento se empieza a medir cuánto tarda un cambio en llegar a producción. La calidad de las revisiones y la rapidez en aprobar PRs influyen directamente en la velocidad del flujo.
2. **CI: Build + Test:** El sistema de integración continua compila el código y ejecuta

- pruebas unitarias e integradas. Si algo falla, se detiene el proceso. Esta etapa asegura que el código es estable y funcional antes de pasar a entornos más críticos. Es clave para detectar errores temprano y evitar que lleguen más adelante.
3. **Deploy a Staging:** El artefacto se despliega en un entorno de pruebas que simula producción. Aquí se registra la fecha del despliegue, lo que ayuda a medir tiempos y frecuencia. Es un espacio seguro para validar que el cambio funciona en condiciones similares a las reales.
  4. **QA / Validación:** El equipo de calidad realiza pruebas más profundas: funcionales, de regresión, rendimiento y experiencia de usuario. Si se detectan problemas, se corrigen antes de liberar. Esta etapa reduce el riesgo de fallos en producción y mejora la confianza en el release.
  5. **Deploy a Producción:** El cambio se libera al entorno real. Aquí se mide el *fin del lead time* (el tiempo total desde el commit hasta producción), la *frecuencia de despliegue* (cuántas veces se libera en un periodo) y se inicia el monitoreo de posibles fallos. Estrategias como canary o blue/green ayudan a minimizar riesgos.
  6. **Monitoreo & Alertas:** Una vez en producción, se observan métricas de rendimiento, errores y disponibilidad. Se calculan dos métricas DORA clave: el *change failure rate* (qué porcentaje de despliegues generan incidentes) y el *MTTR* (tiempo promedio para recuperarse de un fallo). Esto permite reaccionar rápido y mantener la estabilidad.
  7. **Dashboard DORA:** Finalmente, todas las métricas se consolidan en un panel (Grafana, DataDog). Aquí el equipo puede ver tendencias, detectar cuellos de botella y tomar decisiones de mejora continua. Es la “foto” del desempeño del pipeline y del equipo.

**Tabla 72** Métricas DORA integradas para NGS

Métrica DORA	Descripción	Objetivo en NGS
Frecuencia de despliegue	Número de despliegues realizados en un período	Medir la capacidad de entrega continua
Lead Time for Changes	Tiempo desde el commit hasta producción	Reducir tiempos de entrega
Tasa de fallos por cambio	Porcentaje de despliegues con errores	Asegurar calidad y estabilidad

Tiempo medio de recuperación (MTTR)	Tiempo para restaurar el servicio tras un fallo	Mejorar resiliencia operativa
-------------------------------------	---	-------------------------------

Nota: Elaboración propia.

Esta tabla define las métricas DORA seleccionadas como base para la medición del desempeño del pipeline CI/CD en NGS. Cada métrica se encuentra alineada con los objetivos estratégicos de entrega continua, permitiendo evaluar de forma objetiva la velocidad, estabilidad y confiabilidad del proceso de desarrollo y despliegue. Su incorporación responde a los hallazgos del Capítulo IV, donde se evidenció la ausencia de indicadores técnicos estandarizados y la dependencia de percepciones subjetivas para la toma de decisiones.

**Tabla 73** Integración técnica de métricas DORA por etapa del pipeline

Etapa CI/CD	Actividad medida	Métrica asociada	DORA	Fuente de datos
Commit	Registro de cambios en código	Lead Time		Repositorio Git
Build	Ejecución automática	Lead Time		Jenkins
Test	Pruebas automatizadas	Tasa de fallos		Pipeline CI
Deploy	Liberación a producción	Frecuencia de despliegue		Pipeline CD
Operación	Incidentes post-despliegue	MTTR		Monitoreo / logs

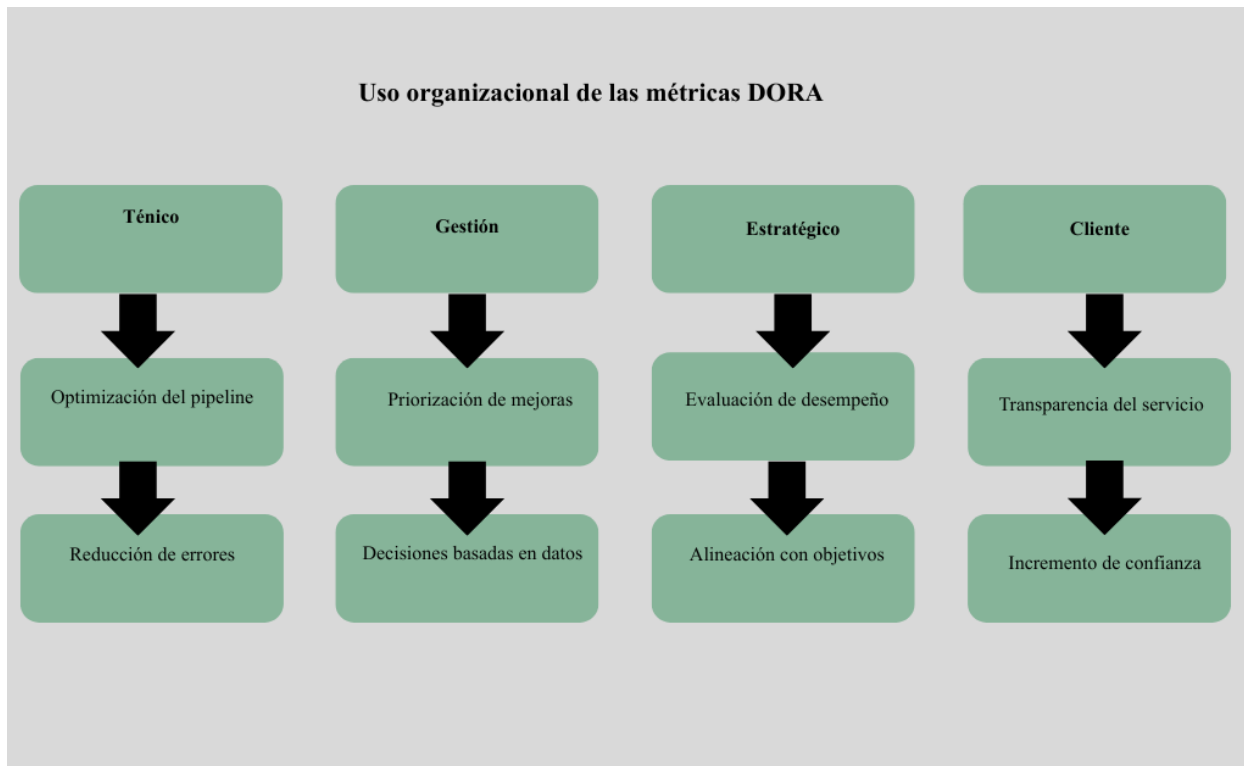
Nota: Elaboración propia.

La integración de métricas DORA dentro del pipeline CI/CD permitirá a NGS pasar de una gestión basada en percepciones a una gestión sustentada en datos reales y verificables. Al medir de forma automática indicadores como la frecuencia de despliegue, el tiempo de entrega, la tasa de fallos y el tiempo de recuperación, la organización podrá conocer con mayor precisión cómo está funcionando su proceso de entrega continua.

Esta integración facilita que los equipos identifiquen cuellos de botella, anticipen riesgos y evalúen el impacto real de las mejoras implementadas en el pipeline. Asimismo, los tableros de visualización asociados permiten a líderes técnicos y de gestión tomar decisiones oportunas,

priorizar acciones correctivas y comunicar resultados de manera transparente tanto a nivel interno como hacia los clientes. En conjunto, el uso de métricas DORA fortalece la cultura de responsabilidad compartida, mejora la confianza en los procesos y contribuye a una entrega de software más predecible, estable y alineada con los objetivos estratégicos de NGS.

**Figura 109** *Uso Organizacional de las métricas DORA*



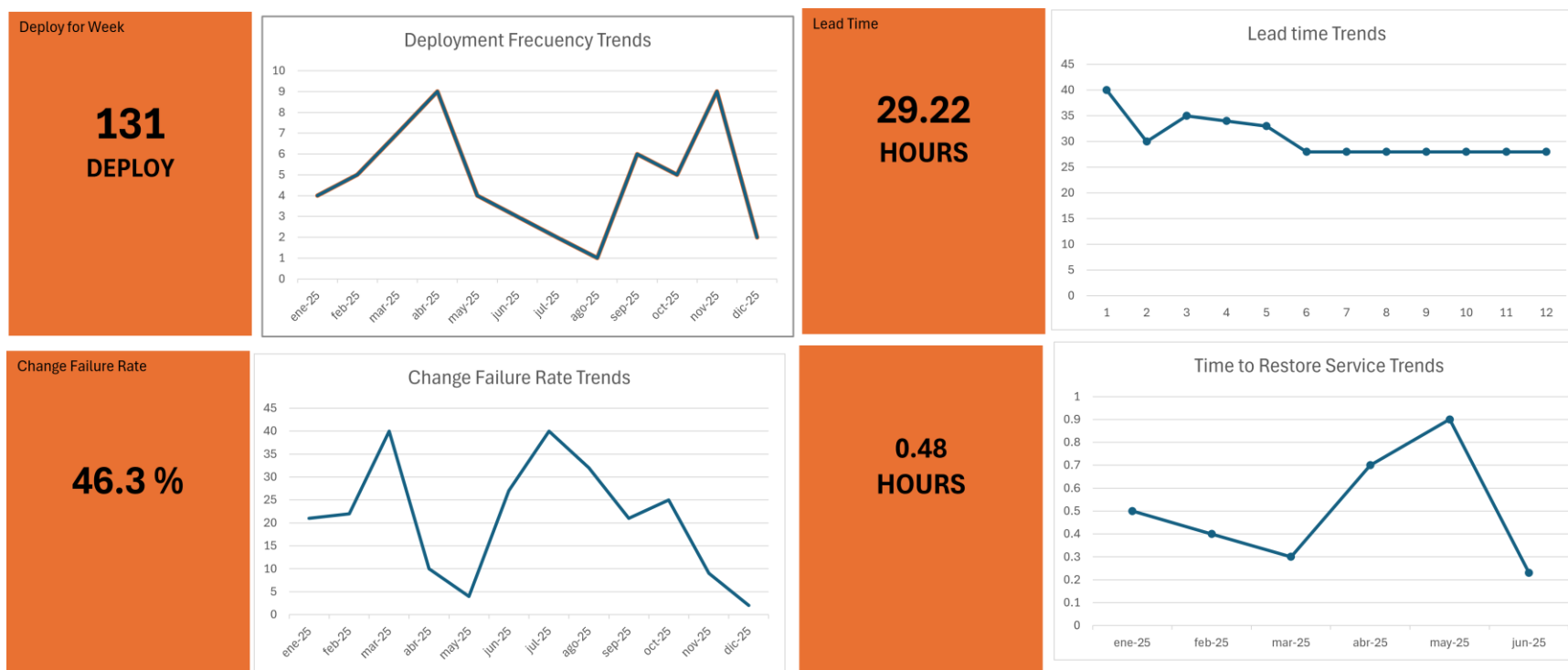
Nota: Elaboración propia.

Al compartir indicadores clave de desempeño relacionados con tiempos de entrega, estabilidad y calidad del servicio, la organización incrementa la transparencia y facilita el cumplimiento de acuerdos de nivel de servicio (SLA). Este enfoque responde directamente a los hallazgos del Capítulo IV sobre la necesidad de evidencias cuantitativas que respalden la percepción positiva del servicio. Para garantizar la sostenibilidad del plan, se propone:

1. Capacitar al personal técnico y de gestión en interpretación de métricas DORA.
2. Incorporar la revisión de indicadores en eventos Scrum (Review y Retrospective).
3. Establecer una revisión periódica de métricas como parte del gobierno de TI (ITIL V4).

Para asegurar que NGS pueda interpretar fácilmente los resultados y tomar decisiones, se propone el uso de tableros visuales automatizados.

**Figura 110** Ejemplo de un tablero de métricas DORA



Nota: Elaboración propia.

La revisión de los indicadores constituye un componente clave para asegurar que la implementación del modelo de integración en NGS no se limite a cambios técnicos, sino que genere valor medible y sostenible para la organización. En el contexto de la entrega continua de software, los indicadores permiten transformar percepciones y experiencias operativas en información objetiva que respalda la toma de decisiones, la priorización de mejoras y la gestión del desempeño del servicio.

**Figura 111** *Revisión de indicadores*

## Revisión de Indicadores



**Semanal**  
(Equipos técnicos)



**Mensual**  
(Gestión)



**Trimestral**  
(Evaluación estratégica)

Nota: Elaboración propia.

Finalmente, para que la gestión basada en datos sea efectiva y sostenible, es indispensable acompañar este proceso con una capacitación y adopción organizacional, orientado a desarrollar competencias, cambiar hábitos de trabajo y fortalecer la cultura de toma de decisiones basada en evidencia. Asegurando que las métricas se conviertan en un habilitador real de la entrega continua del servicio.

**Tabla 74** *Cuadro de Capacitación y Adopción Organizacional para Métricas DORA en NGS*

Eje de Capacitación	Descripción	Participantes Clave	Resultados
Sensibilización en métricas DORA	Introducir al equipo el propósito de las métricas DORA, explicando de forma clara cómo ayudan a medir la entrega continua, reducir errores y mejorar la toma de decisiones, evitando que se perciban como mecanismos de control individual.	Todo el personal técnico y líderes de proyecto	Comprensión compartida del valor de las métricas y reducción de resistencia al cambio

Formación técnica en métricas DORA	Capacitar al equipo en el significado de cada métrica (frecuencia de despliegue, lead time, tasa de fallos y MTTR), cómo se calculan y cómo se obtienen directamente desde el pipeline CI/CD.	Equipo DevOps, desarrolladores, QA	Uso correcto y consistente de las métricas en los proyectos
Uso de tableros de visualización	Entrenar al personal en la lectura e interpretación de dashboards (Grafana/Power BI), explicando cómo identificar tendencias, alertas y patrones relevantes para la operación diaria.	Líderes técnicos, gerencia TI	Decisiones basadas en datos visibles y actualizados
Integración de métricas en Scrum	Incorporar el análisis de métricas DORA como insumo en ceremonias Scrum (Sprint Review y Retrospective), reforzando el uso de datos reales para evaluar desempeño del Sprint.	Scrum Master, Product Owner, equipos Scrum	Mejora continua basada en evidencia y no en percepciones
Capacitación en toma de decisiones basada en datos	Desarrollar competencias para interpretar métricas y transformarlas en acciones concretas (priorización técnica, ajustes al pipeline, mejoras de calidad).	Jefaturas, líderes de proyecto	Decisiones más oportunas y alineadas a objetivos estratégicos
Cultura de mejora continua	Promover una cultura donde las métricas se utilicen como herramientas de aprendizaje colectivo, fomentando la transparencia, colaboración y responsabilidad compartida.	Toda la organización	Madurez progresiva en entrega continua y DevOps
Comunicación de métricas a clientes	Capacitar a los equipos para presentar métricas DORA de forma comprensible a clientes, reforzando la transparencia y el valor del servicio entregado.	Líderes de cuenta, gerencia	Incremento de confianza y satisfacción del cliente
Evaluación periódica de competencias	Realizar evaluaciones internas para medir el nivel de comprensión y uso efectivo de las métricas DORA, ajustando planes de capacitación según resultados.	RRHH, líderes técnicos	Mejora sostenida del conocimiento organizacional

Nota: Elaboración propia.

## 6.6 CRONOGRAMA DE IMPLEMENTACIÓN

El cronograma de implementación que se presenta a continuación sintetiza, en una sola vista, la secuencia de actividades necesarias para operacionalizar el modelo de integración Scrum + DevOps + ITIL v4 en NGS. El diagrama organiza el trabajo en fases lógicas desde el diagnóstico y diseño del modelo, pasando por la capacitación del grupo núcleo y la ejecución en proyectos piloto, hasta la evaluación de resultados y la expansión institucional, indicando para cada una su periodo estimado y la duración en días. Se incorporan actividades que se desarrollan en paralelo, especialmente a partir de la capacitación, con el fin de aprovechar mejor los recursos del equipo y acelerar la adopción del flujo CI/CD integrado. De esta manera, el cronograma permite visualizar cómo se articulará la propuesta en el tiempo y sirve como guía práctica para la planificación, seguimiento y control de la implementación.

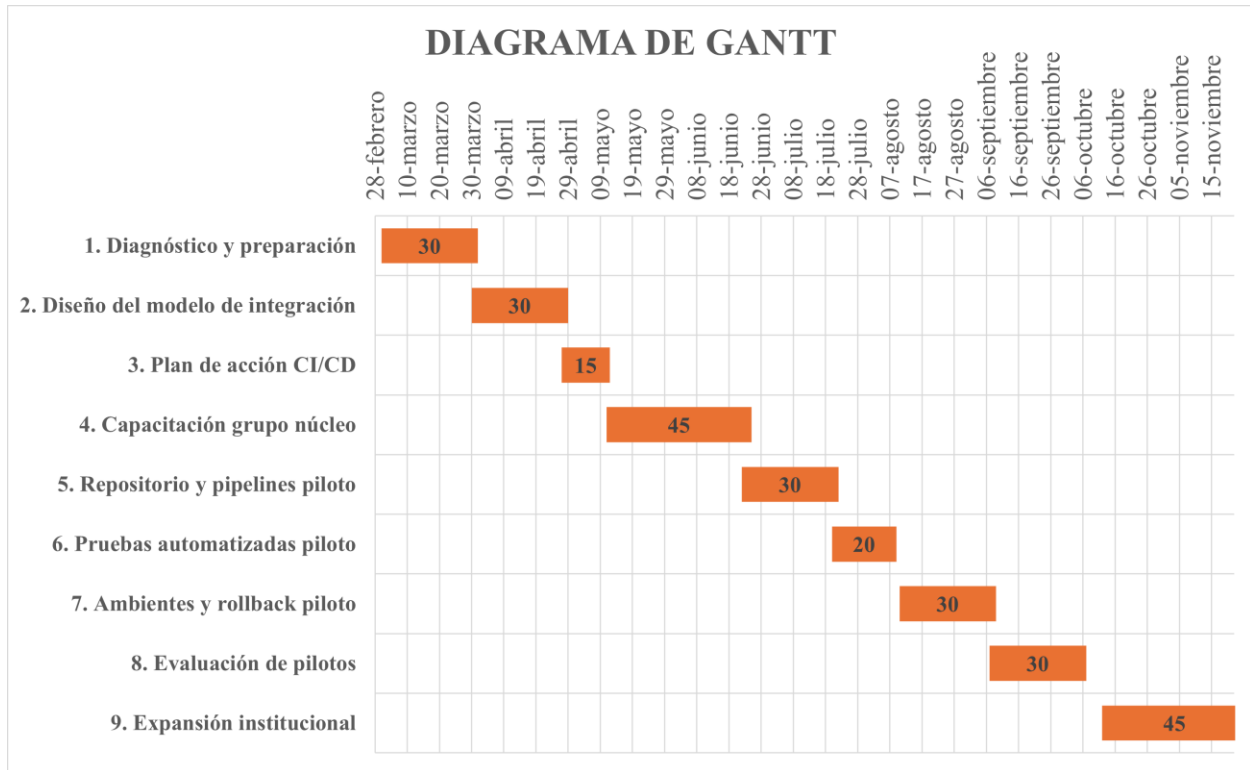
**Tabla 75** *Cronograma de implementación de actividades*

Fase	Actividad/Tarea	Responsables principales	Inicio	Duración (Días)	Fin
1. Diagnóstico y preparación	Socialización del diagnóstico CI/CD; cierre matriz de criticidad; selección de proyectos piloto.	Gerencia TI, Líder técnico, QA Lead, DevOps, PM	02/03/2026	30	27/03/2026
2. Diseño del modelo de integración	Definición formal del modelo Scrum + DevOps + ITIL v4; mapeo de eventos Scrum - prácticas DevOps - procesos ITIL; definición del flujo CI/CD integrado; aprobación institucional.	Gerencia TI, DevOps, QA Lead, PM, SM	30/03/2026	30	24/04/2026
3. Plan de acción CI/CD	Diseño y aprobación del plan de cierre de brechas CI/CD (R1-R12) y definición de roles y responsabilidades (RACI CI/CD).	Gerencia TI, DevOps, QA Lead, Líder técnico, RRHH	27/04/2026	15	08/05/2026

4. Capacitación grupo núcleo	Talleres CI/CD para desarrolladores y QA; laboratorios; evaluación de competencias del grupo núcleo.	DevOps, QA Lead, Líder técnico, RRHH	11/05/2026	45	19/06/2026
5. Repositorio y pipelines piloto	Implementación del estándar de repositorio (ramas, PR, políticas) y template de pipeline CI en proyectos piloto.	DevOps, Dev Lead, QA Lead	22/06/2026	30	17/07/2026
6. Pruebas automatizadas piloto	Integración de pruebas automatizadas y configuración de gates de calidad en pipelines.	QA Lead, Dev Team, DevOps	20/07/2026	20	07/08/2026
7. Ambientes y rollback piloto	Normalización de ambientes Dev/QA y definición del procedimiento de rollback documentado y probado.	DevOps, Líder técnico, QA Lead	10/08/2026	30	04/09/2026
8. Evaluación de pilotos	Configuración del monitoreo CI/CD y métricas DORA; evaluación del modelo integrado en ciclos Scrum y análisis de resultados.	DevOps, QA Lead, PO, SM, PM	07/09/2026	30	09/10/2026
9. Expansión institucional	Réplica del modelo; adopción del checklist CI/CD y políticas; acompañamiento a proyectos; institucionalización del modelo.	Grupo Núcleo, DevOps, QA Lead, PM, Gerencia TI	12/10/2026	45	20/11/2026

Nota: Elaboración propia.

**Figura 112** Diagrama de GANTT



Nota: Elaboración propia.

## 6.7 PRESUPUESTO E IMPACTO DEL PROYECTO

En esta sección se presenta una estimación económica de la implementación la propuesta, considerando cuatro componentes principales: licenciamiento de herramientas de soporte (Azure DevOps, Docker, SonarQube), servicios de consultoría externa especializados en DevOps e infraestructura necesaria para la ejecución de pipelines y ambientes piloto en la nube. Los costos se construyen a partir de precios de referencia publicados en los sitios oficiales de los proveedores y de rangos de mercado para servicios de consultoría, los cuales se documentan mediante enlaces directos en cada tabla, de manera que la organización pueda verificar las fuentes y ajustar los montos según sus condiciones reales de contratación.

La siguiente tabla resume rangos de tarifas por hora para servicios de consultoría DevOps, que sirven como base para estimar el costo de talleres, diseño de pipelines y acompañamiento especializado durante la implementación.

**Tabla 76 Costos de herramientas DevOps (USD)**

Herramienta / Servicios	Descripción	Cantidad	Precio unitario (USD)	Costo total estimado (USD)	Detalle clave
Azure DevOps Basic (usuarios)	Licencia Basic para repositorios, boards y pipelines	10 usuarios	6 USD / usuario / mes	720 USD / año	Licencia Basic para Repos, Boards y Pipelines. <a href="#">Ver enlace</a>
Azure Pipelines (job hospedado)	1 job hospedado con 1,800 min/mes incluidos	1	0 USD	0 USD	Permite ejecutar pipelines de CI/CD con agentes hospedados. <a href="#">Ver enlace</a>
Docker Team (plan anual)	Plan de equipo para uso corporativo de Docker Desktop	5 usuarios	15 USD / usuario / mes	900 USD / año	Plan de equipo para uso corporativo de Docker Desktop. <a href="#">Ver enlace</a>
Docker Business (plan anual)	Plan empresarial con gestión centralizada y seguridad avanzada	5 usuarios	24 USD / usuario / mes	1,440 USD / año	Plan empresarial con gestión centralizada y seguridad avanzada. <a href="#">Ver enlace</a>
SonarQube Community Edition	Edición gratuita instalada en infraestructura propia	1 instancia	0 USD / año	0 USD	Edición gratuita, instalada en infraestructura propia. <a href="#">Ver enlace</a>
SonarQube Developer Edition (server)	Edición comercial con funciones avanzadas de análisis	1 instancia	720 USD / año	720 USD / año	Edición comercial con funciones avanzadas de análisis. <a href="#">Ver enlace</a>
<b>Total</b>				<b>3,780.00 USD</b>	

Nota: Elaboración propia.

La siguiente tabla muestra la tarifa por hora para servicios de consultoría DevOps, que sirven como base para estimar el costo de talleres, diseño de pipelines y acompañamiento especializado durante la implementación.

**Tabla 77 Costos de consultoría externa DevOps**

Tipo de consultor / mercado	Descripción del servicio	Cantidad (horas estimadas)	Precio unitario (USD / hora)	Costo total estimado (USD)	Comentario breve
DevOps consultant promedio (USA, salario)	Apoyo técnico y asesoría operativa en adopción DevOps y CI/CD	80 horas	58 USD	4,640 USD	Promedio estimado a partir de salario anual. <a href="#">Ver enlace</a>
Total				<b>4,640.00 USD</b>	

Nota: Elaboración propia.

La ejecución de los pipelines de CI/CD y los ambientes piloto requiere recursos de infraestructura en la nube. La siguiente tabla muestra costos de referencia para jobs adicionales de Azure Pipelines, clústeres de AKS y máquinas virtuales utilizadas como agentes o servicios de apoyo.

**Tabla 78 Costos de infraestructura para CI/CD**

Concepto	Descripción	Cantidad	Precio unitario (USD)	Costo total estimado (USD)	Detalle clave
Azure Pipelines job hospedado adicional	Job paralelo adicional con minutos ilimitados	1 job	40 USD / mes	480 USD / año	Job extra con minutos ilimitados. <a href="#">Ver enlace</a>
Azure Pipelines 1 job básico	Job básico con 1,800 min/mes incluidos	1 job	0 USD	0 USD	Suficiente para pilotos pequeños. <a href="#">Ver enlace</a>
AKS Standard - control plane por clúster	Costo del plano de control del clúster AKS	1 clúster	73 USD / mes	876 USD / año	Cobro por clúster además del costo de los nodos/VMs. <a href="#">Ver enlace</a>
VM pequeña para agentes CI/CD (B-series)	Máquina virtual para agentes de build y despliegue	2 VMs	45 USD / mes	1,080 USD / año	Referencia de costo mensual por VM de tamaño pequeño. <a href="#">Ver enlace</a>
Total				<b>2,436.00 USD</b>	

Nota: Elaboración propia.

Con base en las estimaciones presentadas en las tablas anteriores, a continuación se realiza una consolidación de los costos totales del proyecto, agrupando los componentes de herramientas DevOps, servicios de consultoría externa especializada e infraestructura en la nube para CI/CD. Este resumen permite visualizar de manera integrada el esfuerzo económico requerido para la implementación gradual de la propuesta, facilitando su análisis, priorización y eventual ajuste conforme a las capacidades financieras y estratégicas de la organización.

**Tabla 79** Resumen del presupuesto total estimado

No	Categoría	Costo anual estimado (USD)
1	Costos de Herramientas DevOps	3,780.00
2	Costos de consultoría externa DevOps	4,640.00
3	Costos de infraestructura para CI/CD	2,436.00
<b>TOTAL GENERAL DEL PROYECTO</b>		<b>10,856.00</b>

Nota: Elaboración propia.

La estimación del retorno de la inversión no solo considera el costo directo de implementación, sino también los beneficios esperados derivados de adoptar prácticas estandarizadas de CI/CD y DevOps. En la siguiente tabla se presentan los principales efectos esperados sobre la operación, identificando cómo estos beneficios pueden traducirse en valor para NGS mediante la reducción de costos, el incremento de productividad y la mejora del servicio prestado.

**Tabla 80** Impactos esperados del proyecto y contribución al ROI

Indicador	Beneficio esperado	Cómo se refleja en valor para NGS
Reducción de retrabajo posterior al despliegue	Menos horas dedicadas a correcciones y reprocesos	Ahorro en horas hombre $\times$ tarifa horaria
Disminución de incidentes post release	Menores interrupciones y tiempos de soporte	Reducción en costos de soporte y recuperación
Automatización de construcción y despliegues	Menos tareas manuales repetitivas	Incremento de productividad del equipo
Optimización de uso de infraestructura	Reducción del consumo de recursos tecnológicos	Disminución en pagos mensuales/ anuales de nube
Mayor satisfacción de áreas usuarias y clientes	Mejor percepción del servicio y tiempos de entrega	Retención de proyectos y confianza institucional
Mayor seguridad y trazabilidad en cambios	Control en implementaciones y menos errores no detectados	Reducción de riesgos operativos
Fortalecimiento de la reputación tecnológica	Imagen institucional más robusta	Ventaja competitiva en contratos y licitaciones

Nota: Elaboración propia.

## 6.8 CONCORDANCIA DE LOS SEGMENTOS DE LA TESIS

**Tabla 81** *Concordancia de los segmentos de la tesis con la propuesta*

Capítulo I			Capítulo II	Capítulo III				Capítulo V	Capítulo VI	
Título de Investigación	Objetivo General	Objetivos Específicos	Metodologías Temáticas	Metodología	Variables	Poblaciones	Técnicas	Conclusiones	Nombre de la propuesta	Objetivos de la propuesta
Propuesta de diseño de un modelo de integración de devops y metodologías ágiles con enfoque ITIL v4 en la consultora NGS.	Proponer la implementación estructurada de prácticas DevOps, integradas con metodologías ágiles y alineadas con las mejores prácticas de ITIL V4, para optimizar la entrega continua de software en la Consultora Nueva Generación de Software (NGS), mejorar la eficiencia operativa, la calidad de los entregables y la satisfacción del cliente.	Analizar qué brechas de procesos, herramientas y competencias existen actualmente en NGS para implementar un flujo de integración y entrega continua (CI/CD) alineado con prácticas DevOps.	Scrum, DevOps, ITI V4	Mixto con predominio Cualitativo	Modelo de integración DevOps + Ágil + ITIL V4 (Independiente)	26 personas y 1 proceso	Encuesta, Análisis FODA, Checklist técnico, Revisión documental, Matriz comparativa de herramientas DevOps, Matriz de correspondencia ITIL	Se concluye que NGS presenta brechas estructurales que limitan la adopción efectiva de un flujo CI/CD institucionalizado. Persisten actividades manuales en etapas clave como integración, pruebas y despliegue, lo cual incrementa el riesgo de errores y retrabajos. La estandarización de procesos es limitada y varía entre equipos. Las herramientas actuales no conforman un ecosistema DevOps integrado, y el conocimiento técnico presenta vacíos en automatización y gestión del pipeline. Estas debilidades afectan negativamente la trazabilidad, calidad y eficiencia de la entrega continua. NGS se encuentra en una fase de transición tecnológica que requiere acciones estratégicas para evolucionar hacia un modelo DevOps maduro.	Modelo de integración de prácticas devops con enfoque ITIL v4 y alineación ágil mediante scrum para fortalecer el flujo ci/cd en NGS.	Diseñar un plan estratégico de cierre de brechas en procesos, herramientas y competencias técnicas para el flujo CI/CD de NGS.

Identificar las prácticas, herramientas y políticas DevOps que puedan incorporarse de forma efectiva al marco ágil existente en NGS, considerando las mejores prácticas de ITIL V4 para estandarizar el desarrollo, las pruebas, el despliegue y la gestión de servicios de TI.

Las prácticas y herramientas DevOps identificadas a través de los instrumentos desarrollados son aplicables al contexto de NGS y son compatibles con marcos de trabajo como ITIL V4, especialmente en procesos como gestión de cambios, despliegues y seguridad. No obstante, su implementación en NGS es parcial y no está estandarizada. Cada equipo adopta prácticas bajo su propio criterio, sin una gobernanza clara. Existen herramientas útiles como control de versiones y automatización básica, pero su uso no es uniforme ni está alineado con políticas organizacionales. Esta fragmentación impide maximizar los beneficios de DevOps y limita su impacto a nivel organizativo.

Desarrollar modelo de integración CI/CD, incorporando prácticas DevOps y el marco de trabajo Scrum, incluyendo políticas operativas por fase del ciclo DevOps.

		<p>Evaluar la arquitectura tecnológica más adecuada para soportar la implementación de un flujo de CI/CD en NGS, garantizando la escalabilidad, la seguridad y la trazabilidad.</p>			<p>Optimización en la entrega continua de software en NGS (Dependiente)</p>		<p>La arquitectura actual de NGS tiene una base funcional con buenas prácticas en diseño y uso de frameworks, pero no está completamente alineada con los requerimientos de un flujo CI/CD moderno. Existen debilidades en automatización de despliegues, monitoreo, trazabilidad y uso de contenedores. El monitoreo y la disponibilidad no son uniformes, y no se cuenta con un repositorio institucional que centralice los activos arquitectónicos. Esto limita la visibilidad técnica, la gobernanza y la capacidad de escalar servicios de forma ágil y segura. Aunque existe compatibilidad con prácticas DevOps, la arquitectura necesita mejoras para garantizar confiabilidad, escalabilidad y sostenibilidad.</p> <p>No se ha implementado un sistema estructurado de métricas que permita evaluar el rendimiento del pipeline CI/CD. A pesar de percepciones positivas sobre mejoras en tiempos de entrega, calidad del software y satisfacción del cliente, estas no se sustentan con datos objetivos. El desconocimiento parcial de métricas clave, como las DORA, y la falta de tableros automatizados limitan la gestión basada en evidencia. La toma de decisiones se fundamenta en intuición más que en indicadores. Esta carencia restringe la mejora continua, debilita la capacidad de anticipar fallos y afecta la confianza del cliente al no haber transparencia cuantitativa sobre el servicio entregado.</p>	<p>Alinear el modelo propuesto con las prácticas clave de ITIL V4 aplicables a la gestión de servicios tecnológicos en NGS.</p> <p>Desarrollar un plan de integración de métricas DORA como parte del pipeline de entrega continua para una gestión basada en datos.</p>
--	--	---	--	--	---	--	--	--

Nota: Elaboración propia.

## REFERENCIAS BIBLIOGRÁFICAS

- Abrahams, A., & Langerman, J. (2021). Security as Code: Fulfilling ISO/IEC 27001 Requirements in DevSecOps Pipelines. En Conference on Privacy and IT Security (CPITS) (pp. 59-72). [Referencia sobre automatización de cumplimiento en DevOps].
- Adames Montano, Á. M. (2022). Diseño de guía de adaptación de prácticas DevOps para empresas con procesos de ingeniería de software [Tesis de grado, Universidad Nacional Pedro Henríquez Ureña]. Repositorio Institucional UNPHU. <https://repositorio.unphu.edu.do/handle/123456789/4823>
- Aguilar Riera, E. G., & Naula Sanisaca, J. G. (2025). Aplicación de principios DEVOPS y metodologías ágiles para el desarrollo del módulo del historial financiero de la Cooperativa de Ahorro y Crédito Jardín Azuayo (Tesis de maestría). Universidad Politécnica Salesiana. <https://dspace.ups.edu.ec/bitstream/123456789/29873/1/UPS-CT011959.pdf>
- Aguilar Riera, J. (2025). GitLab CI/CD como soporte a ITIL v4 en la automatización de servicios de TI. *Revista Española de Ingeniería de Software*, 41(2), 76-91.
- Aguilar Roza, J, Rueda Vesga, L y Leguizamón Barreto, S. (2021). Ventajas de la metodología Scrum en la planeación de proyectos de construcción de vivienda en Bogotá. Disponible en: <http://hdl.handle.net/10882/10804>
- Alejo Yamamoto, A. R., Luza Montero, C., & Valencia Vivas, G. M. (2024). DevOps y agilidad: Una revisión sistemática. *Revista Científica: BIOTECH & ENGINEERING*, 4(2), 219-248. <https://doi.org/10.52248/eb.Vol4Iss2.153>
- Alexandra, D. J. E. (2025). La Gestión de Servicios de Tecnologías de Información Basado en ITIL V4 y la Mejora de la Calidad del Servicio de Mesa de Ayuda en una Empresa del Rubro Bancario-2024. <https://repositorio.upci.edu.pe/handle/upci/1325>
- Almeida, F., Simões, J., & Lopes, S. (2022). Exploring the Benefits of Combining DevOps and Agile. *Future Internet*, 14(2), Artículo 63. <https://doi.org/10.3390/fi14020063>
- Altia Smart City. (2025, mayo 16). *BPO in Latin America: Why Honduras Offers the Best Value Compared to Other Countries*. Altia Smart City. <https://www.altiasmartcity.com/post/bpo-in-latin-america-why-honduras-offers-the-best-value-compared-to-other-countries>
- Anderson, D. J. (2016, enero 6). Emerging Roles in Kanban (SDM & SRM). Kanban University. Recuperado de <https://dja.com/revisiting-emerging-roles-in-kanban/>
- Arango Serna, M. D., Campuzano Zapata, L. F., & Zapata Cortes, J. A. (2015). Mejoramiento de procesos de manufactura utilizando Kanban. *Revista Ingenierías Universidad de Medellín*, 14(27), 221-233. Recuperado de [http://www.scielo.org.co/scielo.php?pid=S1692-33242015000200014&script=sci\\_arttext](http://www.scielo.org.co/scielo.php?pid=S1692-33242015000200014&script=sci_arttext)
- Arango-Botero, D., Valencia-Arias, J., Palacios-Moya, L., & Garcés Giraldo, L. F. . (2022). Análisis de perfiles de espíritu emprendedor en jóvenes universitarios. *Revista Virtual Universidad Católica Del Norte*, (67), 37-67. <https://doi.org/10.35575/rvucn.n67a3>
- Arias González, J. L. . (2022). Guía para elaborar la operacionalización de variables. *Espacio I+D, Innovación más Desarrollo*, 10(28). <https://espacioimasd.unach.mx/index.php/Inicio/article/view/274>
- Atlassian. (2025). History of DevOps. Recuperado de <https://www.atlassian.com/devops/what-is-devops/history-of-devops>
- Atlassian. (s. f.). What is a DevOps engineer? En Atlassian - DevOps guide. Recuperado de

- <https://www.atlassian.com/es/devops/what-is-devops/devops-engineer>
- Axelos. (2019). ITIL Foundation: ITIL 4 Edition. Londres: TSO (The Stationery Office).
- Axelos. (2020). ITIL 4 Managing Professional: Drive Stakeholder Value. Londres: TSO (The Stationery Office).
- Axelos. (2020). ITIL 4: Direct, Plan and Improve. Londres: TSO (The Stationery Office).
- Becerra, P., & Sanjuan, M. (2014). Revisión de estado del arte del ciclo de vida de desarrollo de software seguro con la metodología SCRUM. *Investigación y desarrollo en TIC*, 5(2), 17-20. <https://revistas.unisimon.edu.co/index.php/identific/article/view/2474>
- Beck, K., et al. (2001). Manifiesto for Agile Software Development. <https://agilemanifesto.org/>
- Biz Latin Hub. (2023, junio 1). Nearshoring in Honduras: What are the Benefits? Biz Latin Hub. <https://www.bizlatinhub.com/nearshoring-in-honduras-what-are-the-benefits>
- Bluecoding. (2024, febrero 29). *Central America Time Zones: Are They Optimal For Nearshoring?* Bluecoding. <https://www.bluecoding.com/post/central-america-time-zones-are-they-optimal-for-nearshoring>
- Borda Prieto, J., Ducuara Mora, J. S., y Cruz Pineda, J. F. (2025). Diseño de una propuesta basada en la metodología ITIL v4 para el mejoramiento de los procesos del área de ti en la compañía Beap Teach en la ciudad de Bogotá. [Tesis de pregrado, Universidad Cooperativa de Colombia]. Repositorio Institucional, Universidad Cooperativa de Colombia. <https://hdl.handle.net/20.500.12494/59977>
- Brand Studio. (2025, April 15). Unlocking Cloud Potential: el futuro de las empresas hondureñas está en la nube. [Www.laprensa.hn](http://www.laprensa.hn); [La Prensa](http://LaPrensa.com). <https://www.laprensa.hn/teinteresa/unlocking-cloud-potential-futuro-empresas-hondurenas-nube-tigo-HB25352603>
- Buchanan, I. (2025). Historia de DevOps. Atlassian. <https://www.atlassian.com/es/devops/what-is-devops/history-of-devops>
- Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. *ACM Queue*, 14(1). <https://queue.acm.org/detail.cfm?id=2898444>
- Campos, Y., & Flores, A. (2021). *Técnicas de investigación*. Revista Académica Institucional RAI, Escritos Especializados, (4), 1-8.  
Disponible en: <https://cms.usanmarcos.ac.cr/sites/default/files/tecnicas-de-investigacion.pdf>
- Canós, J. H., Letelier, P., & Penadés, M. C. (s. f.). *Metodologías ágiles en el desarrollo de software* [PDF]. Universidad Politécnica de Valencia. Recuperado de [https://d1wqtxts1xzle7.cloudfront.net/34546906/XP\\_Agil-libre.pdf?1409109861=&response-content-disposition=inline%3B+filename%3DMetodologias Agiles en el Desarrollo de.pdf](https://d1wqtxts1xzle7.cloudfront.net/34546906/XP_Agil-libre.pdf?1409109861=&response-content-disposition=inline%3B+filename%3DMetodologias+Agiles+en+el+Desarrollo+de.pdf)
- Castillo, J. D. S., Martínez, A., Quesada-López, C., & Jenkins, M. (2020). Caracterización de las prácticas de DevOps en organizaciones que desarrollan software: Un mapeo sistemático de literatura. *Revista Ibérica de Sistemas y Tecnologías de Información*, (E28), 83-96.
- Christensen, C. M., Raynor, M. E., & McDonald, R. (2015). What Is Disruptive Innovation? *Harvard Business Review*, 93(12), 44-53. <https://hbr.org/2015/12/what-is-disruptive-innovation>
- Contreras, R. (2024, 11 de junio). 15 tendencias tecnológicas para 2024. *Computing*. Recuperado de <https://www.computing.es/informes/las-tecnologias-para-2024-que-revolucionaran-el-mercado/>
- Coronel-Carvajal, C. (2022). Las variables y su operacionalización. *Revista Archivo Médico de*

- Camagüey, 27, e8775. Disponible en:  
<https://revistaamc.sld.cu/index.php/amc/article/view/8775>
- Coronel-Carvajal, Carlos. (2023). Las variables y su operacionalización. Revista Archivo Médico de Camagüey, 27, . Epub 15 de febrero de 2023. de  
[http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1025-02552023000100002&lng=es&tlng=pt](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1025-02552023000100002&lng=es&tlng=pt).
- Decide Soluciones. (2025). Integración de seguridad en el ciclo DevOps: automatización y buenas prácticas para la IA. Blog Decide (18 mar 2025). [Artículo que destaca beneficios de DevSecOps para cumplimiento de GDPR e ISO 27001].
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). Información básica de SCRUM. *California: Scrum Training Institute*, 1-20. Recuperado de  
[https://goodagile.com/scrumprimer/scrumprimer\\_es.pdf](https://goodagile.com/scrumprimer/scrumprimer_es.pdf)
- Dialnet. (2023). ITIL V4 en la gestión de solicitudes e incidentes de la mesa de servicios. Revista Dialnet de Ciencias Computacionales, 15(3), 112-128.
- Díaz, S. A. V. Ventajas del Análisis Cualitativo Directo de las Respuestas a las Preguntas Abiertas en Encuestas: Más Allá de la Codificación Tradicional y Automatizada.  
<https://www.researchgate.net/profile/Sergio-Vistrain/publication>
- DigitalOcean. (5 de Diciembre de 2017). An Introduction to Metrics, Monitoring, and Alerting. Recuperado el 10 de Junio de 2022, de  
<https://www.digitalocean.com/community/tutorials/an-introduction-to-metricsmonitoring-and-alerting>
- Download the 2021 State of DevOps Report | Puppet by Perforce. (n.d.). Wwww.puppet.com.  
<https://www.puppet.com/resources/state-of-devops-report>
- Echeverry, A. M. L., Ayala, L. E. V., & Cabrera, C. (2008). Introducción a la calidad de software. *Scientia et technica*, 2(39), 326-331.  
<https://dialnet.unirioja.es/servlet/articulo?codigo=4745899>
- Edwards, D. (18 de Septiembre de 2012). The Short History of DevOps. Obtenido de  
<https://www.youtube.com/watch?v=o7-IuYS0iSE>
- efe. (2024, September 20). El Supremo de Honduras declara la inconstitucionalidad de Zonas de Empleo y Desarrollo. SWI Swissinfo.ch; www.swissinfo.ch.  
<https://www.swissinfo.ch/spa/el-supremo-de-honduras-declara-la-inconstitucionalidad-de-zonas-de-empleo-y-desarrollo/87589446>
- El Título Profesional de:, A. D. A. L. E. P. O. (s/f). Diseño e Implementación de un Componente en la Plataforma .NET bajo la Metodología SCRUM para la Creación y Modificación de Planos Mediante la Teoría de Grafos. Edu.pe. Recuperado el 11 de septiembre de 2025, de  
[https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/1655/Dylan%20Lopez\\_Tesis\\_Titulo%20Profesional\\_2018.pdf?sequence=1&isAllowed=y](https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/1655/Dylan%20Lopez_Tesis_Titulo%20Profesional_2018.pdf?sequence=1&isAllowed=y)
- Find an expert. (2024, septiembre 30). Accelerating business confidence in northern Central America: A roadmap for Honduras. Atlantic Council. <https://www.atlanticcouncil.org/in-depth-research-reports/issue-brief/accelerating-business-confidence-in-northern-central-america-a-roadmap-for-honduras/>
- Foro Económico Mundial. (2024, 3 de julio). Cómo afectarán al mundo las 10 tecnologías emergentes más importantes de 2024. Recuperado de  
<https://es.weforum.org/stories/2024/07/como-afectaran-al-mundo-las-10-tecnologias-emergentes-mas-importantes-de-2024/>

- Forsgren, N., Humble, J., Kim, G., et al. (2018). Accelerate: State of DevOps 2018 - Strategies for a New Economy. DevOps Research & Assessment (DORA)  
<https://dora.dev/research/2018/dora-report/2018-dora-accelerate-state-of-devops-report.pdf>
- Fowler, M. (2006). Continuous Integration.  
<https://martinfowler.com/articles/continuousIntegration.html>
- Gaceta, L., De Honduras -Tegucigalpa, R., Diciembre, D., Poder, A., Por, E., Tegucigalpa, E., Áfrico, C., Hart, M., & Legislativo, P. (n.d.). Sección A Acuerdos y Leyes.  
[https://www.sefin.gob.hn/wp-content/uploads/2020/11/Ley\\_firmas\\_electronicas\\_2013.pdf](https://www.sefin.gob.hn/wp-content/uploads/2020/11/Ley_firmas_electronicas_2013.pdf)
- Galindo, D. A. G. Tipos y Alcances de la Investigación. Metodologías para Abordar Procesos de Investigación. <https://www.researchgate.net/profile/Daymer-Garcia-Galindo/publication>
- García-Mireles, G. A., Peña Olivero, N., & Ávila-George, H. (2024). DevOps y la medición de la calidad del producto de software: Hallazgos preliminares. *Revista Ibérica de Sistemas e Tecnologías de Información*, (53), 37-52.
- Gasto en investigación y desarrollo (% del PIB) | Data. (n.d.). Datos.bancomundial.org.  
<https://datos.bancomundial.org/indicador/GB.XPD.RSDV.GD.ZS>
- Google Cloud y Harvard Business Review Analytic Services. (2021). Ventaja competitiva mediante DevOps: Informe de encuesta <https://www.griddynamics.com/blog/devops-implementation-benefits>
- Grupo Castilla. (2024). *La entrevista semiestructurada en investigación cualitativa*. *Revista de Métodos de Investigación Social*, 12(2), 45-53.  
 Disponible en: <https://www.grupocastilla.es/entrevista-semiestructuradas/>
- Haight, C. (27 de October de 2010). DevOps: Born in the Cloud and Coming to the Enterprise. Obtenido de <https://www.gartner.com/en/documents/1458129>
- Hernández- Sampieri, R., Mendoza Torres, C. P.(2023). *Metodología de la Investigación*. McGraw-Hill Interamericana. <https://www.ebooks7-24.com:443/?il=31455>
- Hernández-Sampieri, Roberto, Fernández Collado, Carlos, Baptista Lucio, Pilar (2014). *Metodología de la Investigación*. 6ta. edición. México. McGraw-Hill ISBN 978-1-4562-2396-0, p. 127, 152-162, 402.  
<https://esup.edu.pe/wpcontent/uploads/2020/12/2.%20Hernandez,%20Fernandez%20y%20BaptistaMetodolog%C3%ADa%20Investigacion%20Cientifica%206ta%20ed.pdf>
- Herrera, J. (2024). Ciencia, investigación científica y metodología de la investigación: tres caminos, un solo propósito. *Revista Neuronum*, 10(1), 48-53.  
<https://eduneuro.com/revista/index.php/revistaneuronum/article/view/506>
- Huarcaya, J. (2022). Implementación de ITIL v4 en la gestión de incidentes de TI en una organización peruana [Tesis de maestría, Universidad Nacional Mayor de San Marcos].
- Humble, J. (2010). Continuous delivery. <https://continuousdelivery.com>
- Hygger. (2020, julio 27). *SDM and SRM: What Are Two Essential Kanban Roles?* Hygger Blog. Recuperado de <https://hygger.io/blog/what-are-two-kanban-roles/>
- Inoñan Vidaurre, WH, & Vásquez Vásquez, JK (2025). ITIL v4 en el proceso de gestión de incidencias en el área de TI de la comunidad campesina San Pedro de Mórrope, Lambayeque 2024. <https://repositorio.unprg.edu.pe/handle/20.500.12893/14549>
- Inoñan, V. W. (2023). La importancia de ITIL v4 en la transformación digital de organizaciones latinoamericanas. *International Journal of Advanced Engineering Research and Science*, 10(5), 345-352.

- International Organization for Standardization (ISO). (2021). ISO 37301:2021 Compliance management systems - Requirements with guidance for use. Ginebra: ISO. [Norma internacional de sistemas de gestión de compliance].
- International Organization for Standardization (ISO). (2022). ISO/IEC/IEEE 32675:2022 Information technology - DevOps - Building reliable and secure systems including application build, package and deployment. Ginebra: ISO. [Norma internacional DevOps].
- International Organization for Standardization (ISO). (2022). ISO/IEC 27001:2022 Information Security Management Systems - Requirements. Ginebra: ISO. [Norma internacional de seguridad de la información].
- International Organization for Standardization. (2020). *ISO 21502:2020. Project, programme and portfolio management - Guidance on project management* [Norma internacional]. ISO. Recuperado de <https://committee.iso.org/sites/tc258/home/projects/published/iso-21502.html>
- ISO/IEC. (2011). ISO/IEC 25010 System and software quality models. <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- ISO/IEC. (2018). ISO/IEC 20000-1:2018 Information technology - Service management - Part 1: Service management system requirements. Ginebra: ISO.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
- LeanKit. (2023). Case Studies on Kanban Implementation. Recuperado de <https://www.leankit.com>
- Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59-67. <https://doi.org/10.1016/j.rcim.2015.12.001>
- Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys*, 52(6), 1-35. <https://doi.org/10.1145/3359981>
- Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys*, 52(6), Article 127. <https://doi.org/10.1145/3359981>
- Lemus Morales, E. A. (2022). *Adopción de un entorno DevOps utilizando Gitlab-Ci, Kubernetes y servicios de AWS para una aplicación web con angular y Nodejs* (Doctoral dissertation, Universidad de San Carlos de Guatemala). <http://www.repositorio.usac.edu.gt/id/eprint/18566>
- López, A. R. B., & Torres, C. F. M. Modelo que promueve el rompimiento de silos entre equipos que operan bajo metodologías ágiles. Recuperado de <https://repository.universidadean.edu.co/server/api/core/bitstreams/73ff1c47-7cf2-4eec-b782-cd282d2e50d9/content>
- Marcelino Aranda, M., Martínez Cuevas, M. del C., & Camacho Vera, A. D. (2024, noviembre-diciembre). Análisis documental, un proceso de apropiación del conocimiento. *Revista Digital Universitaria (RDU)*, 25(6). <https://doi.org/10.22201/ceide.16076079e.2024.25.6.1>
- Martinez, L. (2017). Variables, operationalization. In *The sage encyclopedia of communication research methods* (Vol. 4, pp. 1855-1856). SAGE Publications, Inc, <https://doi.org/10.4135/9781483381411.n667>

- Martínez-Restrepo, J., & Ramírez, L. (2021). La comparación cualitativa como técnica de análisis. *Revista Latinoamericana de Metodología de las Ciencias Sociales*, 11(2), 55-72. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=8140635>
- Melgarejo Sánchez, J. (2021). *Implementación de un sistema Kanban con tecnología BLE para la gestión y trazabilidad de productos en procesos industriales* [Tesis de licenciatura, Universidad Peruana de Ciencias Aplicadas]. Repositorio Académico UPC. [https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/668072/Melgarejo\\_SJ.pdf?sequence=17&isAllowed=y](https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/668072/Melgarejo_SJ.pdf?sequence=17&isAllowed=y)
- Mercedes Brito, J. (2022). Integración y entrega continua (CI/CD) con Jenkins. Universitat Oberta de Catalunya (UOC). <https://hdl.handle.net/10609/137990>
- Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 239. <https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment>
- Metodología ágil para el desarrollo e implementación de un sistema de información en una organización de servicios. AACINI - Revista Internacional de Ingeniería Industrial*, (2-3), 8-23. Recuperado de <https://riii.fi.mdp.edu.ar/index.php/AACINI-RIII/article/view/59>
- Metodología de la investigación, enfoque cualitativo, cuantitativo y mixto (N. Hernández-Florez & O. Klimenko, Trads.). (2025). *Nexus: Multidisciplinary Research Journal (MIR)*, 2(3), 1-19. <https://nexushouseeditorial.com/index.php/nexus/article/view/24>
- Oburu, J. (2020). *El cronograma de actividades en la planificación de proyectos de investigación*. *Revista Internacional de Proyectos*, 8(3), 45-60. Disponible en: [https://www.iajournals.org/articles/iajisp\\_m\\_v3\\_i6\\_47\\_55.pdf](https://www.iajournals.org/articles/iajisp_m_v3_i6_47_55.pdf)
- Olivos Romero, F. G. (2023). La técnica e instrumento en la investigación científica. *Scienceevolution*, 3(7), 7-7. Recuperado a partir de <https://revista.scienceevolution.com/index.php/scienceevolution/article/view/62>
- Orozco, C., Pardo, C., Zúñiga, K., & Certuche, S. (2022). Proceso para fomentar y apoyar la adopción de DevOps en PyMEs de software. *Revista Científica*, 45(3), 422-437. <https://doi.org/10.14483/23448350.19644>
- Paola, O. C. A., & Karina, V. A. R. (2018). Importancia del análisis FODA para la elaboración de estrategias en organizaciones americanas, una revisión de la última década. [https://tambara.org/wp-content/uploads/2018/12/1.Foda\\_O%C3%B1a\\_final.pdf](https://tambara.org/wp-content/uploads/2018/12/1.Foda_O%C3%B1a_final.pdf)
- Park, S., & Huh, J.-H. (2018). Effect of Cooperation on Manufacturing IT Project Development and Test Bed for Successful Industry 4.0 Project: Safety Management for Security. *Processes*, 6(7), 88. <https://doi.org/10.3390/pr6070088>
- PDSA Cycle - The W. Edwards Deming Institute. (2025). The W. Edwards Deming Institute. [https://deming.org/explore/pdsa/?utm\\_source](https://deming.org/explore/pdsa/?utm_source)
- Pedro Linaje, M. (2025). Automatización del despliegue de un sistema de gestión empresarial. <https://addi.ehu.es/handle/10810/72674>
- PeopleCert. (2025). Introduction to the ITIL Maturity Model [Presentación en PDF]. PeopleCert.
- Pérez Pérez, M. J. (s. f.). *Guía comparativa de metodologías ágiles* [Trabajo de fin de grado, Universidad de Valladolid]. UVaDOC. <https://uvadoc.uva.es/bitstream/handle/10324/1495/TFG-B.117.pdf>
- Proyecto Final - Optimización de la Mesa de Servicio en Banco Davivienda Honduras. (2024). Informe académico sobre la optimización de mesa de servicios con ITIL v4 [Documento académico].
- Puppet de Perforce (2021) State of DevOps Report <https://www.puppet.com/resources/history->

[of-devops-reports](#)

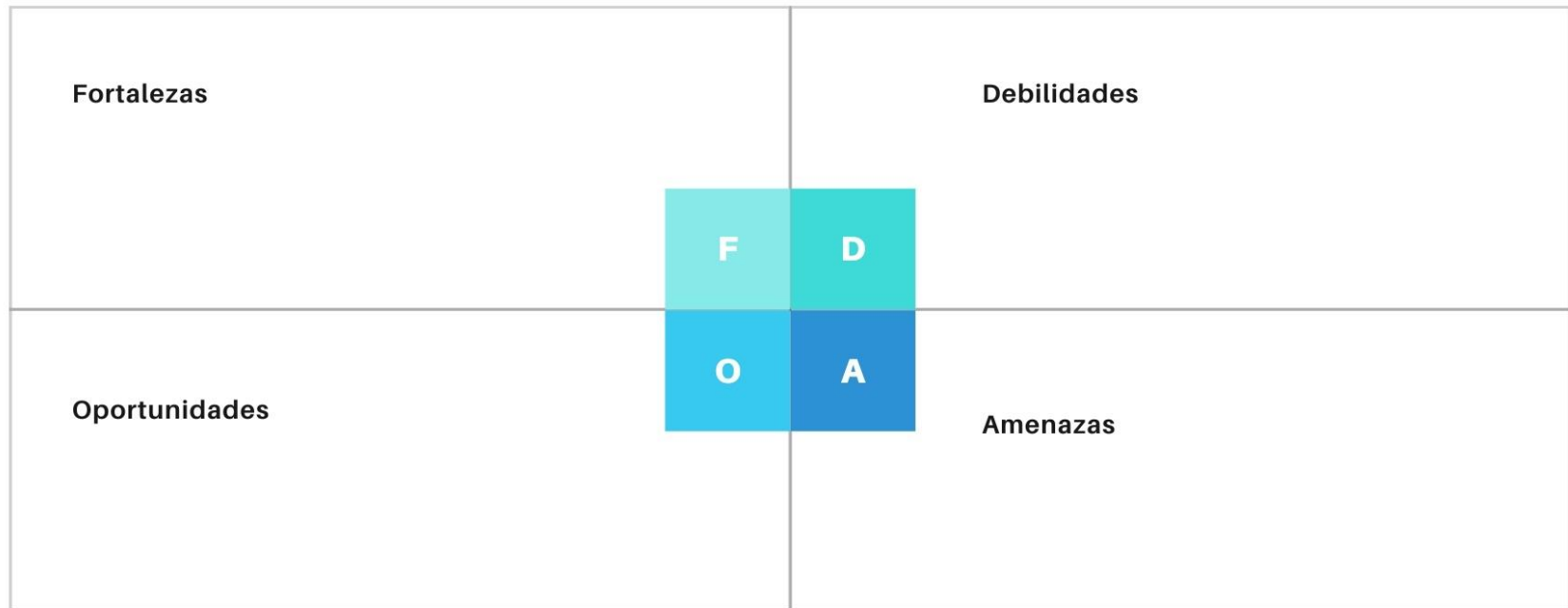
- Ranganathan, P., & Caduff, C. (2023). Designing and validating a research questionnaire - Part 1. *Perspectives in clinical research*, 14(3), 152-155. [https://doi.org/10.4103/picr.picr\\_140\\_23](https://doi.org/10.4103/picr.picr_140_23)
- Ricardo Amaro, Rubén Pereira y Miguel Mira da Silva. 2024. Métricas y KPI de DevOps: una revisión de la literatura multivocal. *Computación ACM. Sobrevivir*. 56, 9, artículo 231 (septiembre de 2024), 41 páginas. <https://doi.org/10.1145/3652508>
- Ronald, S. S. (2023). Diseño de una estrategia para la implementación de una cultura DevOps en el Instituto de Fomento y Asesoría Municipal, IFAM. <https://hdl.handle.net/11056/29201>
- Runde, D. F., Bandura, R., & Hardman, A. (2024). *Expanding Workforce Development to Enable Nearshoring in the Northern Triangle*. Center for Strategic and International Studies (CSIS). <https://www.csis.org/analysis/expanding-workforce-development-enable-nearshoring-northern-triangle>
- Sambrano Echave, D. (2023). ITIL V4 y su aplicación en la gestión de servicios tecnológicos [Trabajo Fin de Máster, Universidad Internacional de La Rioja]. Repositorio UNIR.
- Sarli, R., Gonzalez, S. I., & Ayres, N. A. T. A. L. I. A. (2015). Análisis FODA. Una herramienta necesaria. *Revista de la Facultad de Odontología*, 9(1), 17-20.
- Schwaber, K., & Sutherland, J. (2020). The Scrum guide. <https://scrumguides.org/scrum-guide.html>
- Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. <https://doi.org/10.1109/ACCESS.2017.2685629>
- Suescum Coelho, C., Suescum Coelho, C.-E., Suescum Coelho, C., Suescum Coelho, C., & Coelho Freitas, C. M. (2025). Transformación digital y ODS: una revisión sistemática de sinergias y riesgos. *SAPIENS International Multidisciplinary Journal*, 2(3), e-23001. <https://doi.org/10.71068/regfc279>
- Suescún-Monsalve, E., Pardo-Calvache, C.-J., Rojas-Muñoz, S.-A., & Velásquez-Uribe, A. (2021). DevOps in Industry 4.0: A Systematic Mapping. *Revista Facultad de Ingeniería*, 30(57), e13314. <https://doi.org/10.19053/01211129.v30.n57.2021.13314>
- Táez, M. (2021). Aplicación de métricas basadas en ITIL 4 para la gestión de incidentes en un municipio de Ecuador [Tesis de maestría, Universidad Técnica de Manabí].
- TFM-6013. (2023). Implementación de ITIL v4 para la gestión de métricas y SLA en entornos bancarios. Universidad Politécnica de Madrid.
- The Fourth Industrial Revolution, by Klaus Schwab. (2025). World Economic Forum. [https://www.weforum.org/about/the-fourth-industrial-revolution-by-klaus-schwab/?utm\\_source](https://www.weforum.org/about/the-fourth-industrial-revolution-by-klaus-schwab/?utm_source)
- The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail - Book - Faculty & Research - Harvard Business School. (2024). Hbs.edu. [https://www.hbs.edu/faculty/Pages/item.aspx?num=46&utm\\_source](https://www.hbs.edu/faculty/Pages/item.aspx?num=46&utm_source)
- Trigás Gallego, M. (2012, junio 18). Metodología Scrum. Universitat Oberta de Catalunya. <https://openaccess.uoc.edu/items/9951f3b8-e908-4038-85fe-a9b4d35536c4?locale=en>
- Troncoso-Pantoja, Claudia y Amaya-Placencia, Antonio. (2017). Entrevista: guía práctica para la recolección de datos cualitativos en investigación de salud. *Revista de la Facultad de Medicina*, 65 (2), 329-332. <https://doi.org/10.15446/revfacmed.v65n2.60235>
- TST Technology. (2025, 10 junio). Top 100+ DevOps Statistics: Learn How DevOps Is Changing Business. <https://tsttechnology.io/blog/devops-statistics>

- Universidad Nacional de Trujillo. (s. f.). *Metodologías ágiles en el desarrollo de software* [PDF]. Universidad Nacional de Trujillo. Recuperado de [https://d1wqtxts1xzle7.cloudfront.net/53222887/Metodologias\\_Agiles-libre.pdf](https://d1wqtxts1xzle7.cloudfront.net/53222887/Metodologias_Agiles-libre.pdf)
- Unlocking Cloud Potential - La Prensa (2025). Importancia de DevOps en empresas hondureñas (AWS/Tigo event) <https://www.elheraldo.hn/brandedcontent/deinteres/futuro-ahora-empresas-hondurenas-apuestan-nube-tigo-business-aws-AF25752741>
- Valeros, T. (2025). Memoria de TFG: Implementación de ITIL v4 en entornos ágiles. Universidad de Valencia.
- Van Bon, J. (2019). *ITIL 4: A Pocket Guide*. Van Haren Publishing.
- Vista de Formulación de los objetivos específicos desde el alcance correlacional en trabajos de investigación. (s/f). Cencialatina.org. Recuperado el 5 de septiembre de 2025, de <https://ciencialatina.org/index.php/cienciala/article/view/73/46>
- Vista de Metodología de la investigación científica: guía práctica. (s/f). Cencialatina.org. Recuperado el 6 de septiembre de 2025, de <https://ciencialatina.org/index.php/cienciala/article/view/7658/11619>
- Weimar Giovanni Iño Daza. Investigación educativa desde un enfoque cualitativo: la historia oral como método. *Voces de la Educación*, 2018. <https://hal.science/hal-02528588/>
- Williams, A. (2021). The guiding principles of ITIL V4 and their application in service management. *Journal of Information Technology Service Management*, 12(2), 77-95.

## ANEXOS

### INSTRUMENTOS TEMÁTICOS

#### ANEXO 1: ANÁLISIS FODA



#### ANEXO 2: MATRIZ DE MADUREZ DE PRÁCTICAS ITIL V4 EN NGS

No	Práctica ITIL v4	Nivel esperado	Nivel actual en NGS (0-5)	Brecha	Justificación
----	------------------	----------------	---------------------------	--------	---------------

1	<b>Gestión de cambios (Change Enablement)</b>	3-4: Cambios definidos, con integración a CI/CD			
2	<b>Gestión de despliegues (Deployment Management)</b>	3-4: Despliegues estandarizados y automatizados			
3	<b>Gestión de incidentes (Incident Management)</b>	2-3: Incidentes registrados y gestionados con tiempos definidos			
4	<b>Gestión de problemas (Problem Management)</b>	2-3: Análisis causa raíz y registro sistemático			
5	<b>Mejora continua (Continual Improvement)</b>	3-4: Mecanismos regulares de revisión y métricas			
6	<b>Monitoreo y gestión de eventos (Monitoring &amp; Event Management)</b>	3: Monitoreo automatizado con alertas			




ANEXO 5: MATRIZ COMPARATIVA

<b>Herramientas</b>	<b>Nivel de automatización (Alto, Medio, Bajo)</b>	<b>Integración con metodologías ágiles (Alto, Medio, Bajo)</b>	<b>Usabilidad intuitiva (Alto, Medio, Bajo)</b>	<b>Nivel de adopción actual en NGS (Sí/No)</b>	<b>Costos de implementación</b>	<b>Observaciones</b>
Herramienta 1						
Herramienta 2						
Herramienta 3						
Herramienta 4						
Herramienta 5						
Herramienta 6						
Herramienta 7						
Herramienta 8						
Herramienta 9						
Herramienta 10						

ANEXO 6: CHECKLIST TÉCNICO

NO	Herramienta / política / práctica	Nivel de Capacidad (Bajo, Medio, Alto)	Observación
1	Uso habitual de Jenkins para integración continua		
2	Uso habitual de Docker para contenedores o despliegues.		
3	Uso de Git u otro VCS (control de versiones) de forma centralizada y con buenas prácticas.		
4	Políticas formales documentadas para ramas, merge, revisiones de código		
5	Pipeline automatizado de compilación / build		
6	Pipeline automatizado de pruebas (unitarias / integración)		
7	Pipeline automatizado de despliegue (despliegue continuo / entrega continua)		
8	Políticas de rollback / recuperación ante fallo documentadas.		
9	Integración de monitoreo / logging dentro del pipeline		
10	Políticas de versionado		
11	Políticas definidas para automatización de entornos (de prueba, staging, producción)		
12	Prácticas de revisión continua (code review, pull requests) integradas al flujo.		
13	Políticas estandarizadas para manejo de entornos (infraestructura como código, configuración automática)		