



Evaluación del impacto de un plan de acción basado en prácticas DevOps en el área de producto de Hello Insight

Daniel Montenegro Cordero

Proyecto de grado entregado para obtener el título de
Magister en Ingeniería de Software

Dirigida por
Ph.D. Luisa Rincón

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Maestría en Ingeniería de Software
Santiago de Cali
26 de Enero de 2026

Agradecimientos

Este trabajo de maestría es el resultado de un proceso exigente y profundamente formativo, que no habría sido posible sin el apoyo de las personas e instituciones que me rodean. A todas ellas les expreso mi gratitud por impulsarme y acompañarme en las distintas etapas de este camino académico.

En primer lugar, agradezco a mi familia por su respaldo constante y por la paciencia durante los momentos de mayor carga, que me permitió mantenerme enfocado cuando más lo necesitaba para culminar este proceso. De igual manera, agradezco a mis amigos y compañeros de la maestría por su acompañamiento; con ellos compartí discusiones, retroalimentación y aprendizajes que enriquecieron mi perspectiva y contribuyeron en este proceso formativo.

Agradezco de manera especial a mi directora de tesis, *Luisa Rincón*, por su orientación rigurosa, sus observaciones oportunas y su acompañamiento durante el desarrollo del trabajo, y a la *Pontificia Universidad Javeriana - Cali* por brindarme el espacio académico y las condiciones formativas que hicieron posible desarrollar este proyecto.

De manera particular, agradezco a *Hello Insight* por brindar el contexto y la apertura para desarrollar este trabajo, así como a los colegas que participaron en entrevistas, mediciones y actividades. Su colaboración, disposición y compromiso con el proyecto fueron fundamentales para llevar a cabo la intervención.

Resumen

Este trabajo presenta la evaluación de la madurez DevOps en el área de producto de *Hello Insight* mediante un ciclo de investigación–acción. Se estableció una línea base aplicando el modelo de capacidades de Molina (2022) y se complementó el diagnóstico con métricas DORA para caracterizar la velocidad y estabilidad de entrega. Con base en los hallazgos, se identificaron y priorizaron capacidades clasificadas como oportunidades de mejora y se diseñó un plan de acción de cuatro semanas orientado a incorporar prácticas DevOps mediante cambios recomendados por el modelo de capacidades. Posteriormente, se aplicó una segunda medición para comparar el estado antes y después de la intervención, calculando cambios por capacidad, por dimensión y a nivel global. Los resultados evidenciaron un incremento del puntaje total de 2.44 a 3.57 (+1.13), sin retrocesos, y un desplazamiento general hacia categorías superiores del modelo, incluyendo mejoras notables en la dimensión de cultura. Finalmente, se discuten implicaciones prácticas, amenazas a la validez, lecciones aprendidas y una ruta de trabajo futuro para orientar el siguiente ciclo de mejora.

Palabras Clave: DevOps, madurez DevOps, investigación-acción, modelo de capacidades, métricas DORA.

Abstract

This work presents an assessment of DevOps maturity in the Product area of *Hello Insight* through an action-research cycle. A baseline was established using the capability model proposed by Molina (2022), and the diagnosis was complemented with DORA metrics to characterize delivery speed and stability. Based on the findings, capabilities classified as improvement opportunities were identified and prioritized, and a four-week action plan was designed to incorporate DevOps practices through changes recommended by the capability model. A second measurement was then applied to compare the state before and after the intervention, computing changes at the capability, dimension, and overall levels. Results show an increase in the total score from 2.44 to 3.57 (+1.13), with no regressions, and a general shift toward higher maturity categories, including notable improvements in the Culture dimension. Finally, practical implications, threats to validity, lessons learned, and a future work roadmap are discussed to guide the next improvement cycle.

Keywords: DevOps, DevOps maturity, action research, capability model, DORA metrics.

Índice general

Agradecimientos	7
1. Introducción	1
1.1. Introducción	1
1.2. Definición del problema	2
1.2.1. Planteamiento del problema	2
1.2.2. Formulación del problema	3
1.3. Objetivos del proyecto	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Alcance	4
1.4.1. Limitaciones	5
1.5. Justificación	6
1.6. Metodología de la investigación	7
1.6.1. Diagnóstico	8
1.6.2. Planificación	8
1.6.3. Acción	9
1.6.4. Evaluación	9
1.6.5. Aprendizaje	9
1.7. Resultados esperados	9
2. Marco de referencia	11
2.1. Marco teórico de referencia y antecedentes	11
2.1.1. Bases Teóricas	11
2.1.2. Estado del Arte	18
3. Diagnóstico y Planificación	23
3.1. Modelo de Capacidades	23
3.1.1. Estructura del Modelo de Capacidades de Molina (2022)	24
3.1.2. Estructura de las Preguntas Modelo de Capacidades de Molina (2022)	24
3.1.3. Opciones de Respuesta	25
3.1.4. Plan de Acción	25
3.2. Implementación Encuesta de Modelo de Capacidades	25
3.2.1. Preparación	26
3.2.2. Ejecución	27
3.2.3. Resultados	27
3.3. Métricas DORA 2025 (Enero - Agosto) en Hello Insight	36

3.3.1. HI-API	36
3.3.2. HI-Frontend	38
3.3.3. HI-Analytics	40
3.4. Socialización de Resultados y Priorización de Capacidades	41
3.5. Plan de Acción	44
3.5.1. Capacidades Priorizadas	44
3.5.2. Actividades	46
3.5.3. Definición de Actividades	47
3.5.4. Resumen consolidado	49
3.6. Cronograma de Actividades	49
3.7. Resumen del capítulo	50
4. Acción y Evaluación	53
4.1. Ejecución Plan de Acción	54
4.1.1. Metodología de ejecución	54
4.1.2. Ejecución: Dimensión - Capacidad	54
4.1.3. Cierre del plan de acción	59
4.1.4. Resultados preliminares del plan de acción	59
4.2. Evaluación - Reassessment(Segunda aplicación del modelo)	60
4.2.1. Preparación	60
4.2.2. Ejecución	61
4.2.3. Resultados	61
4.2.4. Impacto en el negocio de Hello Insight	72
4.3. Resumen del capítulo	73
5. Conclusiones	75
5.1. Conclusiones	75
5.2. Amenazas a la validez	76
5.3. Trabajos futuros	77
5.4. Lecciones aprendidas	78
5.5. Cierre	79
A. Capacidades con oportunidades de mejora y plan de acción sugerido por el artefacto de Molina	81
B. Definición detallada de las actividades del plan de acción definitivo	85
C. Formato para ADR en Hello Insight	89
D. Script de procesamiento de respuestas	91
Bibliografía	93

Índice de figuras

1.1. Ciclo iterativo de la metodología Investigación-acción. Imagen tomada de Adolph et al. (2020)	7
2.1. Relación entre capacidades, prácticas y resultados. Imagen tomada de Amaro et al. (2023)	14
3.1. Estructura del Modelo de Capacidades propuesto por Molina. Imagen tomada de Molina (2022).	24
3.2. Estructura de la Pregunta del Modelo de Capacidades. Imagen tomada de Molina (2022).	25
3.3. Limpieza de las respuestas. Imagen propia, tomada de la herramienta Colab.	28
3.4. Nota general del nivel de madurez DevOps en <i>Hello Insight</i>	31
3.5. Nota por dimensión del nivel de madurez DevOps en <i>Hello Insight</i>	32
3.6. Conteo de clasificaciones por dimensión en <i>Hello Insight</i>	33
3.7. <i>Whiteboard</i> para lluvia de ideas.	46
3.8. Resultado de la lluvia de ideas.	47
3.9. Cronograma de Actividades Plan de Acción	50
4.1. Diagrama General Procesos de CI/CD en Hello Insight	55
4.2. Diagrama de flujo (Alluvial): transición de capacidades entre categorías (diagnóstico → evaluación)	66
4.3. Distribución de capacidades según clasificación durante el diagnóstico y la evaluación	67
4.4. Comparativo por dimensión (barras): diagnóstico vs. evaluación	68
4.5. Radar por dimensión (promedio): diagnóstico vs. evaluación	69
4.6. Gráfico comparativo del resultado total del diagnóstico vs. la evaluación	70

Índice de tablas

2.1. Prácticas DevOps y su relación con atributos de calidad del software	14
3.1. Métricas recolectadas de la encuesta del modelo de capacidades	27
3.2. Clasificación de resultados según el benchmark del modelo de capacidades	29
3.3. Resultados obtenidos tras el diagnóstico	31
3.4. Resumen: Promedio y distribución porcentual por categoría	33
3.5. Capacidades clasificadas como oportunidades de mejora	34
3.6. Capacidades clasificadas como aceptables	34
3.7. Capacidades clasificadas como fortalezas	35
3.8. Frecuencia de despliegue HI-API (2025)	36
3.9. Tiempo de recuperación HI-API	37
3.10. Lead Time for Changes HI-API	37
3.11. Change Failure Rate HI-API	37
3.12. Frecuencia de despliegue HI-FRONTEND (2025)	38
3.13. Tiempo de recuperación HI-FRONTEND	38
3.14. Lead Time for Changes HI-FRONTEND	39
3.15. Change Failure Rate HI-FRONTEND	39
3.16. Frecuencia de despliegue HI-ANALYTICS (2025)	40
3.17. Tiempo de recuperación HI-ANALYTICS	40
3.18. Lead Time for Changes HI-ANALYTICS	40
3.19. Change Failure Rate HI-ANALYTICS	41
3.20. Comparación de priorización preliminar (Jefe de Producto vs. Equipo) y priorización final consensuada	43
3.21. Capacidades priorizadas para el plan de acción	44
3.22. Resumen consolidado de esfuerzo estimado por dimensión para 4 semanas.	49
4.1. Resultados obtenidos tras el <i>reassessment</i>	64
4.2. Comparación de resultados por ítem entre diagnóstico y <i>reassessment</i> , con delta por pregunta	65
4.3. Comparación diagnóstico vs. <i>reassessment</i> para capacidades priorizadas (prioridad 1)	65
4.4. Capacidades seleccionadas como prioridad 1 para el siguiente ciclo de mejora	72
A.1. Capacidades con oportunidades de mejora y su plan de acción	83

Introducción

1.1. Introducción

En los últimos años, el enfoque DevOps ha cobrado gran relevancia en la industria del software por su capacidad de acelerar la entrega de productos de calidad mediante la integración de equipos de desarrollo y operaciones, promoviendo prácticas de automatización, colaboración continua y mejora constante. Sin embargo, la implementación efectiva de DevOps no depende únicamente de incorporar herramientas tecnológicas, sino que requiere transformar la cultura organizacional, los procesos internos y la arquitectura técnica.

Diversas organizaciones han reportado casos de éxito en la adopción de prácticas DevOps. Empresas como *Netflix*, *Amazon* y *Google* han logrado reducir significativamente los tiempos de entrega, aumentar la confiabilidad de sus sistemas y mejorar la eficiencia operativa gracias a la aplicación de este enfoque. Estos resultados, documentados en estudios como los informes anuales de DORA y el libro *Accelerate* de [Forsgren et al.](#), evidencian el valor de adoptar prácticas DevOps desde una perspectiva integral, y sus casos de éxito han permitido que otras empresas de tecnología tengan en cuenta este enfoque para sus procesos de desarrollo de software.

En respuesta a la necesidad de las organizaciones de comprender y orientar dicha adopción, se han desarrollado modelos y marcos conceptuales que permiten evaluar el nivel de madurez DevOps en las organizaciones, entre los que se destacan CALMS, DORA y el modelo propuesto en *Accelerate*. A partir de estos aportes, trabajos como el de [Molina \(2022\)](#) han consolidado artefactos de software más estructurados que facilitan el diagnóstico de las capacidades DevOps en múltiples dimensiones, ofreciendo un punto de partida para establecer estrategias de adopción adaptadas al contexto de cada organización.

Hello Insight es una empresa pequeña, constituida por 30 empleados aproximadamente, dedicada principalmente al desarrollo de una única plataforma web enfocada en la generación de reportes a partir de la evaluación del impacto de programas educativos y comunitarios en las capacidades esenciales que los jóvenes necesitan para prosperar, como habilidades sociales, autogestión, entre otras, y el desarrollo positivo de los jóvenes (*Positive Youth Development, PYD*)¹. Su estructura organizacional se divide en tres grandes ramas: la primera rama, liderada por la Jefa de Operaciones (*Chief Operations Officer*), encargada del área de ventas y éxito del cliente (*Member Success*); la segunda rama, liderada por la Jefa Ejecutiva (*Chief Executive Leader*), con algunas modificaciones recientes, encargada de las áreas de mercadeo y recursos humanos; y finalmente, la tercera rama, controlada por el Jefe de Producto (*Chief Product Officer*) responsable del área de producto.

¹¿Qué medimos en Hello Insight? Disponible en <https://helloinsight.org/what-we-measure>

El área de producto de *Hello Insight* incluye todo lo relacionado con el desarrollo y operación del producto tecnológico. Es el área encargada de liderar tanto la implementación de nuevas funcionalidades como el mantenimiento y operación de la plataforma. Algunas prácticas DevOps han ayudado a soportar estos procesos, pero a pesar de ya tener algunas de ellas, *Hello Insight* carecía de una estrategia formal de implementación y no contaba con un entendimiento acerca de sus procesos actuales. Esto generaba dificultades para consolidar una cultura de mejora continua, afectaba la eficiencia en los despliegues y limitaba la sostenibilidad de los procesos de desarrollo y entrega de software.

Para abordar este problema se aplicó el modelo de capacidades propuesto por Molina (2022) donde se pudo obtener el nivel de madurez actual de la empresa y unas recomendaciones de buenas prácticas de DevOps, y se complementó con las métricas DORA y la validación del juicio experto del equipo de desarrollo. Teniendo en cuenta todo lo anterior, se diseñó y ejecutó un plan de acción durante un mes, enfocado en fortalecer dimensiones clave de los procesos DevOps en el área de producto, como la automatización (Entrega Continua) y la monitorización (Gestión Lean y Monitorización)

Dicho plan se implementó en el área de producto de la empresa, con un enfoque orientado a fortalecer sus procesos de desarrollo y entrega de software. Como cierre, se realizó un nuevo análisis del nivel de madurez alcanzado por la organización, en el que se resaltan los cambios obtenidos. Estos resultados sentaron las bases para una adopción más sostenible de DevOps en la organización y generaron aprendizajes que servirán como guía para futuros desarrollos y escalamiento de sus procesos.

1.2. Definición del problema

1.2.1. Planteamiento del problema

Hello Insight ofrece una plataforma web especializada en la recolección de datos para el análisis de programas implementados por organizaciones sin ánimo de lucro en diversos entornos, como colegios y universidades en los Estados Unidos. Su producto permite la generación automática de reportes basados en factores calculados para medir distintas capacidades relacionadas con el aprendizaje y el desarrollo positivo de los jóvenes. Entre sus funcionalidades, destaca un panel de gestión de encuestas, que facilita el monitoreo del progreso de los grupos, la descarga de datos y el acceso a diversas opciones de gestión de reportes. Gracias a estas características, la plataforma se ha convertido en una herramienta integral para que las organizaciones sin ánimo de lucro evalúen sus programas académicos, maximicen su impacto en las comunidades donde operan y puedan obtener más recursos.

El área de producto en *Hello Insight* es la responsable del funcionamiento y la evolución de la plataforma web. Actualmente está conformado por un Jefe de Producto (*CPO*); un Gerente de Proyecto (*Project Manager, PM*); un diseñador; un Analista de Producto; un recientemente agregado equipo de investigación ("data mining") y un equipo de desarrollo compuesto por un líder técnico y tres desarrolladores. Dado el tamaño reducido del equipo, es común que los miembros desempeñen

múltiples funciones. Por ejemplo, el líder técnico no solo coordina al equipo de desarrollo, sino que también se encarga de la administración de bases de datos y brinda soporte a la operación.

Con la evolución de la plataforma desarrollada por *Hello Insight* y su creciente impacto en el sector, los procesos, la tecnología y los miembros del equipo del área de producto también han cambiado. Tras la última reestructuración, el equipo actual ha heredado prácticas y herramientas previamente adoptadas por líderes técnicos que hoy en día no forman parte de la organización. Algunas de estas prácticas están alineadas con prácticas DevOps, con el objetivo de apoyar el proceso de desarrollo.

Al asumir estos procesos legados, el área de producto intentó continuar con la adopción de DevOps como soporte para el proceso de desarrollo de software. Sin embargo, diversas deficiencias en su implementación dificultaron la sostenibilidad de estos procesos, afectando procesos clave como los despliegues, las pruebas y la cultura DevOps. Dentro de estas deficiencias se encuentran la falta de documentación, la ausencia de mantenimiento de las librerías, la rápida obsolescencia de algunas tecnologías, junto con la falta de una comunidad activa que las respalde. Como consecuencia, los miembros del área han tenido que recurrir a más procesos manuales, lo que ha impactado negativamente en la adopción de la metodología y, asimismo, en el proceso de desarrollo de software.

A pesar de los intentos previos de adoptar DevOps, la falta de un enfoque estructurado ha dificultado su consolidación en *Hello Insight*. La ausencia de una estrategia clara para la gestión de procesos, herramientas y cultura organizacional ha limitado la capacidad del equipo para garantizar una entrega continua, eficiente y sostenible.

En este contexto, este proyecto busca evaluar el estado actual de madurez de DevOps en el área de producto de la empresa *Hello Insight*, identificar las áreas más críticas y proponer un plan de acción que fortalezca tanto la adopción de sus prácticas como la consolidación de su cultura. El objetivo es no solo impulsar las prácticas DevOps en los procesos del ciclo de desarrollo de software, sino también iniciar un proceso que ayude a fomentar una cultura organizacional que garantice la sostenibilidad del producto y de las tecnologías que lo respaldan a largo plazo.

1.2.2. Formulación del problema

¿Cómo evaluar la madurez de DevOps en el área de producto de la empresa *Hello Insight* para identificar oportunidades de mejora y fortalecer su adopción?

- ¿Cómo aplicar el modelo de capacidades propuesto por Molina (2022) para evaluar la madurez de DevOps en el área de producto de *Hello Insight*?
- ¿Cuál es el nivel actual de madurez de DevOps en el área de producto de *Hello Insight*?
- ¿Cuáles son las áreas críticas que requieren mejoras para fortalecer la adopción de DevOps en el área de producto de *Hello Insight*?
- ¿Qué prácticas deben incluirse en un plan de acción para trabajar en los puntos críticos detectados?
- ¿Cuáles son los resultados obtenidos a partir de la ejecución del plan de acción?

1.3. Objetivos del proyecto

1.3.1. Objetivo general

Evaluar el impacto de un plan de acción basado en prácticas DevOps, en el área de producto de *Hello Insight*, a partir de la comparación del nivel de madurez antes y después de su implementación, utilizando el modelo de capacidades de [Molina \(2022\)](#).

1.3.2. Objetivos específicos

- Analizar el modelo de capacidades propuesto por [Molina \(2022\)](#) y ajustarlo al contexto del área de producto de *Hello Insight* para su aplicación en la evaluación de la madurez de DevOps.
- Aplicar el modelo de capacidades adaptado para evaluar el nivel de madurez de DevOps en el área de producto.
- Identificar y priorizar oportunidades de mejora con base en los resultados obtenidos y en las necesidades estratégicas del área de producto.
- Diseñar e implementar un plan de acción, con una duración aproximada de 4 semanas, enfocado en impulsar la adopción de prácticas DevOps en el área de producto de *Hello Insight*.
- Comparar el estado del área de producto antes y después de la implementación del plan de acción, con el fin de analizar su impacto en el nivel de madurez de DevOps en el área de producto *Hello Insight*.

1.4. Alcance

Este proyecto se enfoca en evaluar la madurez de DevOps en el área de producto de la empresa *Hello Insight* utilizando el modelo de capacidades propuesto por [Molina \(2022\)](#), con el fin de diseñar e implementar un plan de acción que permita impulsar la adopción de las prácticas DevOps en los procesos de desarrollo de software. El alcance del trabajo incluye:

- Realizar una revisión del instrumento implementado y determinar su aplicabilidad para el área de producto de la empresa *Hello Insight*.
- Definir una estrategia para aplicar el modelo de evaluación en el área de producto, que incluye:
 - Selección de los participantes a quienes se aplicará el cuestionario propuesto por [Molina \(2022\)](#).
 - Definición de métodos y herramientas a utilizar para la recolección de datos.
 - Selección de las métricas clave que serán empleadas en el análisis de los resultados.

- Recolectar información sobre las métricas actualmente disponibles en el área de producto, identificando cuáles se encuentran implementadas y cuáles se requieren para la evaluación.
- Realizar el diagnóstico de la madurez de los procesos DevOps aplicando el modelo de evaluación con la estrategia definida.
- Establecer los criterios para priorizar las oportunidades de mejora identificadas en el diagnóstico, asegurando su alineación con los objetivos estratégicos de la empresa y las capacidades disponibles.
- Clasificar y agrupar las oportunidades de mejora según su impacto y viabilidad, diferenciando entre mejoras de corto, mediano y largo plazo.
- Validar las prioridades definidas con el equipo del área de producto de *Hello Insight*.
- Documentar las prioridades establecidas.
- Elaborar un plan de acción para la implementación de mejoras en DevOps, basado en las prioridades identificadas y en buenas prácticas ajustadas desde la literatura.
- Implementar el plan de acción con un alcance limitado a las prioridades previamente definidas.
- Documentar los resultados obtenidos durante la ejecución del plan de acción.
- Realizar nuevamente el diagnóstico de madurez después de la implementación del plan de acción.
- Comparar métricas antes y después de la ejecución para evaluar el impacto.
- Medir el impacto del plan de acción en términos de crecimiento en los niveles de madurez DevOps.
- Elaborar un informe final con el diagnóstico, las recomendaciones y las buenas prácticas implementadas, incluyendo el impacto del plan de acción y propuestas de trabajo futuro.

1.4.1. Limitaciones

1. Limitaciones Técnicas:

- La evaluación se aplicó exclusivamente sobre las herramientas, procesos y entornos ya existentes en el área de producto de *Hello Insight*.

2. Limitaciones de Alcance:

- La evaluación se centró en los procesos de desarrollo del área de producto, por lo que no se abordaron procesos de otras áreas de la empresa.

3. Limitaciones de Tiempo:

- El proyecto tuvo una duración total de seis meses, de los cuales un mes estuvo destinado a la implementación del plan de acción.
- Las mejoras que se ejecutaron dentro del marco del proyecto fueron el resultado de una priorización basada en la evaluación. No se implementaron todas las buenas prácticas recomendadas, sino únicamente aquellas definidas como prioritarias.
- La ejecución del proyecto requirió la participación de diferentes miembros del equipo con distintos roles y responsabilidades. La disponibilidad de estos colaboradores pudo verse limitada, lo que afectó los tiempos de implementación y recolección de información.

4. Limitaciones de Recursos:

- Se priorizó el uso, evaluación y mejora de las herramientas actuales con las que contaba el equipo.
- En caso de que surgiera la necesidad de adoptar nuevas herramientas o tecnologías, no se consideraron aquellas cuyo costo superara el presupuesto disponible del equipo.

Este proyecto proporcionó un diagnóstico detallado del nivel de madurez de DevOps en el área de producto de *Hello Insight* y un conjunto de recomendaciones prácticas para su mejora. Además, generó aprendizajes que podrán servir de base para futuras iniciativas de fortalecimiento de DevOps en la empresa.

1.5. Justificación

La adopción de prácticas DevOps ha demostrado ser un factor clave en la mejora de la calidad del software, la entrega continua y la consolidación de una cultura de colaboración en las organizaciones. Según [Forsgren et al. \(2018\)](#), las organizaciones que implementan DevOps alcanzan una mayor estabilidad en sus despliegues y una reducción significativa en los tiempos de entrega, lo que impacta positivamente en la eficiencia operativa. En equipos pequeños de desarrollo, la integración de DevOps permite optimizar el ciclo de vida del software, minimizando los riesgos asociados a la implementación y asegurando un flujo de trabajo más ágil y predecible.

En el caso de empresas pequeñas como *Hello Insight*, la adopción de DevOps cobra especial importancia debido a la falta de estructuras organizativas definidas y personal altamente especializado. Según [Pérez-Mergarejo et al. \(2014\)](#), las pequeñas y medianas empresas (PYMES) a menudo enfrentan barreras en la implementación de modelos de madurez debido a su tamaño y disponibilidad de recursos. DevOps proporciona una solución eficiente al permitir la estandarización de procesos, la integración continua y la automatización, lo que facilita la escalabilidad sin comprometer la calidad del software. Sin una estrategia de madurez DevOps, las pequeñas empresas pueden experimentar retrasos en la entrega de productos, dificultades para adaptarse a los cambios del mercado y mayores costos operativos.

Este proyecto buscó evaluar el estado actual de madurez de DevOps en el área de producto de *Hello Insight* e implementar un plan de acción que permitiera impulsar la adopción de DevOps,

comparando los resultados obtenidos antes y después a través del modelo de capacidades propuesto por Molina (2022). Se estableció un diagnóstico detallado y un conjunto de prácticas orientadas a promover mejoras, cuya implementación fue medida en función de los resultados obtenidos, permitiendo valorar el impacto de la intervención y tomar decisiones informadas para un eventual crecimiento en los niveles de madurez DevOps en el área de producto.

1.6. Metodología de la investigación

El presente trabajo de grado adopta la metodología de investigación-acción (Action Research), un enfoque ampliamente reconocido en ingeniería de software cuando se busca actuar sobre un contexto organizacional para generar mejoras concretas y, simultáneamente, construir conocimiento científico. Este enfoque fue formalizado en el ámbito de la ingeniería de software por Adolph et al. (2020), quienes lo presentan como una estrategia especialmente útil para comprender y abordar procesos complejos en entornos reales. En particular, destacan su relevancia para analizar fenómenos relacionados con los procesos de desarrollo, la cultura organizacional y la adopción de nuevas prácticas dentro de equipos técnicos. Por lo tanto, esta metodología resulta apropiada para los objetivos de este proyecto, ya que permite evaluar el nivel de madurez actual y, a partir del diagnóstico, elaborar un plan de acción que facilite la mejora en la adopción de prácticas DevOps en una organización como *Hello Insight*.



Figura 1.1: Ciclo iterativo de la metodología Investigación-acción. Imagen tomada de Adolph et al. (2020)

La investigación-acción se basa en un ciclo iterativo de diagnóstico, planificación, acción e interpretación de resultados, como se puede evidenciar en la Fig. 1.1, las cuales se explican a continuación junto con las actividades específicas que se llevaron a cabo en la empresa *Hello Insight*. Aprovechando el carácter cíclico de esta metodología, se realizaron revisiones periódicas con la directora del proyecto cada 15 días, con el fin de analizar avances, validar decisiones y realizar los ajustes necesarios en función de los hallazgos obtenidos.

1.6.1. Diagnóstico

En esta fase, se realizó un análisis del estado actual de las prácticas relacionadas con la entrega de software dentro del área de producto de *Hello Insight*. Para ello, se aplicará un modelo de evaluación de madurez DevOps propuesto por Molina (2022), el cual está diseñado para identificar el nivel de adopción en dimensiones como cultura organizacional, automatización, integración continua, entrega continua y gestión de infraestructura.

Las actividades a desarrollar incluyen:

- Analizar el modelo de madurez basado en capacidades planteado por Molina (2022).
- Adaptar el modelo a la estructura y necesidades de la empresa *Hello Insight*.
- Realizar una revisión detallada de los procesos de desarrollo y operación existentes dentro de la organización.
- Obtener métricas clave relacionadas con el ciclo de vida del software (frecuencia de despliegue, tiempos de entrega, tasa de fallos, entre otros), que servirán como insumo para el diagnóstico.
- Seleccionar a los miembros del equipo de producto y operaciones que participarán en el proceso de evaluación.
- Aplicar el instrumento de diagnóstico adaptado a los participantes seleccionados, con el fin de establecer una línea base del nivel de madurez DevOps actual.
- Determinar el estado de madurez de los procesos de entrega de software del área de producto de la empresa *Hello Insight*.

1.6.2. Planificación

Con base en los resultados obtenidos durante el diagnóstico, se diseñará un plan de acción de un mes orientado a impulsar la adopción de prácticas DevOps en el área de producto de *Hello Insight*. Este plan incluirá iniciativas priorizadas de acuerdo con:

- El nivel de madurez encontrado.
- La factibilidad técnica y organizacional.
- Los recursos disponibles.

1.6.3. Acción

Durante esta fase se ejecutarán las actividades definidas en el plan de acción.

1.6.4. Evaluación

Una vez ejecutadas las acciones, se procederá a evaluar su efectividad mediante una segunda aplicación del instrumento de madurez DevOps, además de entrevistas y análisis comparativo frente a la línea base establecida en el diagnóstico.

1.6.5. Aprendizaje

Finalmente, se realizará una reflexión crítica sobre el proceso completo, destacando los aprendizajes organizacionales, las lecciones metodológicas y las posibilidades de replicabilidad en otros contextos. Esta reflexión permitirá:

- Documentar lecciones aprendidas para futuras iniciativas DevOps en *Hello Insight*.
- Consolidar conocimiento útil para otras organizaciones con desafíos similares.
- Identificar trabajos a futuro.

1.7. Resultados esperados

- Establecer una línea base del nivel de madurez DevOps del área de producto de *Hello Insight* mediante la aplicación del modelo de capacidades de Molina (2022), con resultados por capacidad, por dimensión y un puntaje global.
- Identificar y priorizar oportunidades de mejora a partir de los resultados del diagnóstico y las necesidades estratégicas del equipo.
- Diseñar e implementar un plan de acción de aproximadamente cuatro semanas con seguimiento entre brechas detectadas, actividades ejecutadas y evidencias generadas.
- Implementar mejoras verificables en prácticas DevOps (por ejemplo, automatización en CI/CD, criterios de calidad y documentación mínima) y centralizarlas en un repositorio de referencia.
- Aplicar una segunda medición del instrumento para comparar el estado antes y después del plan de acción, calculando deltas por capacidad y cambios agregados por dimensión.
- Evidenciar el comportamiento de las capacidades luego del plan de acción: impacto directo sobre dimensiones priorizadas y posibles efectos secundarios.
- Documentar lecciones aprendidas, limitaciones y recomendaciones para un siguiente ciclo de mejora, dejando un enfoque replicable para evaluaciones futuras.

Marco de referencia

2.1. Marco teórico de referencia y antecedentes

2.1.1. Bases Teóricas

Esta sección presenta los fundamentos teóricos que sustentan el proyecto de grado, proporcionando un marco conceptual para analizar y mejorar los procesos de DevOps en el área de producto de *Hello Insight*. El propósito es diagnosticar el nivel de madurez actual de la organización en cuanto a la adopción de DevOps, identificar oportunidades de mejora y estructurar un plan de acción que permita fortalecer estas prácticas. Asimismo, se busca establecer los criterios necesarios para evaluar el impacto de las mejoras implementadas.

2.1.1.1. DevOps

Definición

DevOps se define como un conjunto de principios y prácticas que facilitan una mejor comunicación y colaboración entre los actores involucrados, con el objetivo de especificar, desarrollar y operar productos y servicios de software y sistemas, promoviendo además la mejora continua en todas las etapas del ciclo de vida del desarrollo (Committee, 2021). El término DevOps surge de la combinación de “Desarrollo” (Development, Dev) y “Operaciones” (Operations, Ops), y hace referencia a un enfoque colaborativo en el que ambos equipos trabajan de manera integrada para eliminar las barreras organizacionales que impiden el flujo libre y eficiente de información entre equipos, departamentos o áreas, basándose en un cambio cultural (Amaro et al., 2023).

Principios

En *The DevOps Handbook*, Kim et al. (2021) describen tres principios fundamentales que sustentan DevOps, basados en el concepto de Las Tres Vías (The Three Ways). Estos principios guían la implementación de DevOps en organizaciones:

1. **La Primera Vía - Flujo (Flow)**: Se centra en optimizar el flujo de trabajo desde el desarrollo hasta la entrega en producción. Para lograrlo, se busca reducir cuellos de botella, eliminar desperdicios y fomentar la automatización en el flujo de trabajo (*pipeline*) de despliegue. Herramientas como integración continua (CI) y entrega continua (CD) juegan un papel clave (Kim et al., 2021).

2. **La Segunda Vía - Retroalimentación (Feedback):** Enfatiza la creación de mecanismos de retroalimentación rápida entre los equipos de desarrollo, operaciones y seguridad. Esto permite identificar y corregir errores de manera temprana, mejorando la calidad del software. Se utilizan estrategias como monitoreo en tiempo real y pruebas automatizadas (Kim et al., 2021).
3. **La Tercera Vía - Aprendizaje y Experimentación Continua:** Fomenta una cultura de mejora continua, experimentación segura y aprendizaje organizacional. Se incentiva la toma de riesgos controlados y la automatización de tareas repetitivas para permitir a los equipos centrarse en la innovación (Kim et al., 2021).

Capacidades DevOps

Una capacidad, en el contexto de DevOps, se entiende como el conjunto de habilidades organizacionales, técnicas y culturales que pueden ser desarrolladas, medidas y optimizadas, y que permiten mejorar continuamente el rendimiento en la entrega de software, así como alcanzar mejores resultados a nivel organizacional (Forsgren et al., 2018). Por su parte, en Amaro et al. (2023) definen una capacidad DevOps como la habilidad para realizar una acción dentro del dominio DevOps o como la cualidad o estado de ser capaz de hacerlo. En ese mismo trabajo, los autores señalan que estas capacidades se componen de habilidades combinadas que los miembros de una organización han acumulado y desarrollado con el tiempo. Por ejemplo, las capacidades tecnológicas en DevOps se relacionan con el conocimiento y las habilidades —técnicas, de gestión e institucionales— que permiten a las organizaciones utilizar el equipamiento y la tecnología de forma eficiente. Estas capacidades pueden pertenecer a individuos, equipos o a la organización en su conjunto, y su mejora contribuye a la madurez y efectividad de la implementación de DevOps (Forsgren et al., 2018).

Las capacidades identificadas en Forsgren et al. (2018) se agrupan en cinco grandes categorías, cada una relacionada con aspectos clave en el mejoramiento del desempeño de la entrega de software:

1. Capacidades de Entrega Continua

- Control de versiones
- Automatización de despliegues
- Integración continua
- Desarrollo basado en trunk (rama principal)
- Automatización de pruebas
- Gestión de datos de prueba
- Seguridad anticipada (shift left en seguridad)
- Entrega continua

2. Capacidades de Arquitectura

- Arquitectura desacoplada
- Equipos empoderados

3. Capacidades de Producto y Proceso

- Retroalimentación del cliente
- Flujo de valor
- Trabajo en pequeños lotes
- Experimentación por parte del equipo

4. Capacidades de Gestión Lean y Monitoreo

- Procesos de aprobación de cambios
- Monitoreo
- Notificación proactiva
- Límites de trabajo en proceso (WIP)
- Visualización del trabajo

5. Capacidades Culturales

- Cultura organizacional de Westrum
- Apoyo al aprendizaje
- Colaboración entre equipos
- Satisfacción laboral
- Liderazgo transformacional

Prácticas DevOps

DevOps cuenta con un conjunto de prácticas que integran las distintas etapas del ciclo de vida del desarrollo ágil de software, con el objetivo de optimizar la entrega continua de productos y mejorar la eficiencia organizacional. Desde la perspectiva de [Forsgren et al. \(2018\)](#), las prácticas DevOps consisten en acciones concretas —como la automatización de despliegues, el control de versiones o el uso de integración continua— que permiten obtener resultados tangibles en la entrega de software y que pueden ser definidas, medidas y mejoradas sistemáticamente. Estas prácticas pueden ser realizadas tanto por individuos como por equipos —como el de ingeniería— y, además, pueden actuar como habilitadoras de una capacidad específica. Mientras que las capacidades se abordan desde una perspectiva evaluativa, las prácticas se entienden desde un enfoque práctico, centrado en la ejecución. En la figura 2.1, podemos ver la relación que existe entre capacidades, prácticas y resultados según [Amaro et al. \(2023\)](#), en la que las prácticas permiten activar capacidades, y estas, a su vez, influyen en la obtención de resultados, como una mayor eficiencia o una mejor calidad del software.

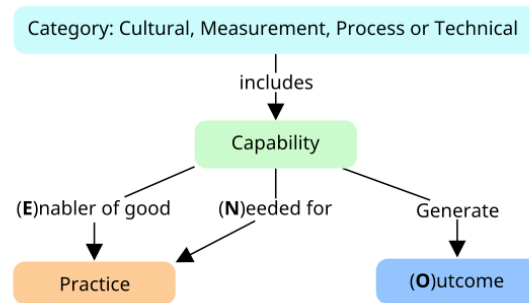


Figura 2.1: Relación entre capacidades, prácticas y resultados. Imagen tomada de [Amaro et al. \(2023\)](#)

En el artículo de [Kato et al. \(2022\)](#), se identifican y clasifican un conjunto de prácticas DevOps clave que inciden directamente en la calidad del software dentro de un entorno de integración y entrega continua (CI/CD). Los autores destacan que estas prácticas no solo automatizan tareas críticas del proceso de desarrollo, sino que también permiten garantizar características de calidad como confiabilidad, mantenibilidad, eficiencia del rendimiento, seguridad y portabilidad. En la Tabla 2.1 se muestra la lista de prácticas de DevOps más importantes mencionadas en [Kato et al. \(2022\)](#).

Práctica DevOps	Características de calidad asociadas
Reglas de codificación	Mantenibilidad
Análisis estático	Fiabilidad, Mantenibilidad
Pruebas unitarias	Funcionalidad, Fiabilidad
Pruebas de integración	Funcionalidad, Fiabilidad
Pruebas de regresión	Funcionalidad, Fiabilidad
Pruebas E2E	Funcionalidad, Usabilidad, Fiabilidad
Medición del tiempo de procesamiento del pipeline	Rendimiento
Pruebas de instalación	Portabilidad
Pruebas de seguridad (DAST)	Seguridad
Pruebas de rendimiento	Rendimiento
Gestión de resultados de pruebas	Fiabilidad, Mantenibilidad

Tabla 2.1: Prácticas DevOps y su relación con atributos de calidad del software

CALMS

El modelo CALMS es un marco conceptual que proporciona los principios clave para la adopción

de DevOps en las organizaciones. Este enfoque combina aspectos culturales, procesos automatizados, metodologías ágiles y medición continua para optimizar la entrega de software y mejorar la colaboración entre equipos (Atlassian, nd).

El acrónimo C.A.L.M.S se define como:

1. **Cultura colaborativa (Culture):** Promueve la colaboración y comunicación entre equipos de desarrollo, operaciones y seguridad, fomentando una mentalidad de responsabilidad compartida.
2. **Automatización de procesos (Automation):** Reduce el trabajo manual mediante la automatización de procesos como la integración, las pruebas, el despliegue y el monitoreo.
3. **Dirección o manejo Lean:** Se enfoca en la mejora continua, eliminando desperdicios y optimizando el flujo de trabajo para acelerar la entrega de software.
4. **Medir, monitoreo constante (Measurement):** Recolecta datos sobre rendimiento, calidad y estabilidad para evaluar el impacto de los cambios y garantizar mejoras constantes.
5. **Colaboración (Share):** Fomenta la transparencia y el intercambio de conocimientos entre equipos, fortaleciendo la colaboración y el aprendizaje organizacional.

Métricas

Forsgren et al. (2018), en *Accelerate: Building and Scaling High Performing Technology Organizations*, identificaron cuatro métricas clave para evaluar el desempeño de los equipos de DevOps. Desde entonces, estas métricas se han consolidado como el marco principal para medir tanto la velocidad como la estabilidad en la entrega de software.

De acuerdo con el informe *DevOps Research and Assessment (DORA) (2023)*, las métricas continúan siendo fundamentales, aunque el enfoque se ha ampliado para incluir también dimensiones organizacionales y culturales que impactan directamente en el rendimiento. Las métricas actualizadas son:

1. **Lead Time for Changes (Tiempo de entrega de cambios):** Mide el tiempo que transcurre desde que un cambio en el código es implementado hasta que está disponible en producción. Un menor tiempo refleja mayor agilidad para responder a las necesidades del negocio.
2. **Deployment Frequency (Frecuencia de despliegue):** Indica la cantidad de veces que un equipo libera software en producción. Los equipos de alto desempeño despliegan múltiples veces al día, mientras que los de menor rendimiento lo hacen con menor frecuencia (por ejemplo, mensual o menos).
3. **Failed Deployment Recovery Time (Tiempo de recuperación de un despliegue fallido):** A partir del reporte DORA 2023, la métrica tradicional *MTTR (Mean Time to Restore)* fue reemplazada por esta métrica más específica, que mide el tiempo que tarda

un equipo en restaurar el servicio después de un despliegue fallido. El cambio busca mayor precisión al enfocarse exclusivamente en fallos derivados de despliegues y no en cualquier incidente de producción.

4. **Change Failure Rate (Tasa de fallos en cambios)**: Representa el porcentaje de despliegues que generan fallos en producción y requieren intervención correctiva, como un rollback o una corrección urgente.

Forsgren et al. (2018) concluyen que medir y optimizar estas métricas permite a las organizaciones mejorar la eficiencia en la entrega de software sin comprometer la estabilidad. Estas métricas han sido adoptadas ampliamente en la industria como indicadores clave de rendimiento (*Key Performance Indicators - KPI*) para DevOps.

2.1.1.2. Modelos de Madurez en DevOps

Un modelo de madurez es un marco conceptual diseñado para evaluar y mejorar las capacidades, procesos o prácticas dentro de una organización. Estos modelos establecen fases secuenciales de evolución, lo que permite diagnosticar el estado actual, definir objetivos estratégicos y trazar un plan de mejora estructurado.

En el contexto de DevOps, los modelos de madurez son utilizados para medir el grado de adopción de sus principios y prácticas dentro de una organización. Según la revisión sistemática de literatura realizada por Jayakody and Wijayanayake (2024), estos modelos permiten determinar el nivel de implementación de DevOps en una empresa y establecer un plan de acción para su optimización. Sin embargo, no existe un estándar único en la industria, y diferentes enfoques han sido desarrollados para evaluar la madurez en este ámbito.

La implementación de un modelo de madurez permitirá establecer prioridades y diseñar acciones que optimicen los procesos de desarrollo de software. Cada uno de estos modelos permite evaluar aspectos clave como la automatización, la calidad del software, la colaboración entre equipos y la cultura organizacional, proporcionando una estructura metodológica para la mejora continua en la implementación de DevOps.

Modelos Empresariales de Madurez DevOps

Las organizaciones han desarrollado diversos modelos de madurez en DevOps con el propósito de evaluar, estructurar y mejorar la implementación de sus prácticas. Estos modelos empresariales buscan proporcionar un marco de referencia para guiar a las compañías en la adopción de DevOps, asegurando que los procesos de desarrollo y operación evolucionen de manera estructurada y alineada con los objetivos estratégicos del negocio. Las grandes empresas han desarrollado modelos de madurez en DevOps para evaluar y mejorar su adopción de prácticas DevOps. Estos modelos ofrecen un marco estructurado para optimizar procesos, automatización y cultura organizacional.

1. IBM DevOps Maturity Model

- Estructurado en cinco niveles: Inicial, Gestionado, Definido, Medido y Optimizado.
- Se enfoca en la automatización, monitoreo y mejora continua para alcanzar una cultura DevOps madura (Jayakody and Wijayanayake, 2024).

2. Capgemini DevOps Maturity Model

- Evalúa tres dimensiones: Personas (colaboración y cultura), Procesos (metodologías ágiles) y Herramientas (automatización y monitoreo).
- Permite a las empresas definir estrategias de adopción según sus necesidades (Jayakody and Wijayanayake, 2024).

3. Eficode DevOps Maturity Model

- Propone cinco niveles: Caos, Repetible, Definido, Medible y Optimizado.
- Se enfoca en CI/CD, pruebas automatizadas y métricas clave como Lead Time y Change Failure Rate (Jayakody and Wijayanayake, 2024).

4. xMatters Atlassian DevOps Maturity Model

- Evalúa la madurez en cinco áreas: Cultura y alineación, Operaciones y soporte, Entrega continua, Diseño y arquitectura y Pruebas y validación.
- Facilita la implementación de prácticas DevOps en la gestión del ciclo de vida del software (Jayakody and Wijayanayake, 2024).

CMMI para DevOps

Adaptación del modelo *Capability Maturity Model Integration (CMMI)*, ampliamente utilizado en la industria del software para medir la madurez de los procesos organizacionales (CMMI Institute, 2025). Facilita la mejora continua al proporcionar un enfoque estructurado para la evaluación de procesos y la implementación de mejoras incrementales. Aplicado en el contexto DevOps, este modelo permite a las organizaciones medir su capacidad en términos de integración, automatización y cultura colaborativa.

Está estructurado en cinco niveles de madurez:

- Inicial
- Gestionado
- Definido
- Cuantitativamente Gestionado
- Optimizado

DSOMM de OWASP

DevSecOps Maturity Model (DSOMM) es un marco desarrollado por OWASP que evalúa la madurez de DevSecOps con un enfoque específico en la seguridad (OWASP Foundation, 2025). Se centra en dimensiones clave como la protección de código, seguridad en infraestructura, monitoreo continuo y cultura de seguridad dentro de los equipos de desarrollo. Además, integra prácticas de seguridad en cada fase del ciclo de vida del desarrollo de software, asegurando que la seguridad no sea un aspecto aislado sino una parte fundamental del proceso DevOps.

Propone cuatro niveles de madurez:

- Básico
- Intermedio
- Avanzado
- Experto

2.1.2. Estado del Arte

2.1.2.1. Modelo de Capacidades de Molina (2022)

El modelo de capacidades propuesto por Molina (2022) es un artefacto de medición diseñado para evaluar la adopción de prácticas DevOps en organizaciones, con un enfoque en la mejora del rendimiento en la entrega de software. Se basa en un marco de referencia que identifica los elementos clave para la implementación efectiva de DevOps y ayuda a establecer estrategias de mejora.

Estructura del Modelo

El modelo se compone de los siguientes elementos:

- 5 dimensiones: Representan áreas clave dentro de DevOps.
- 24 capacidades: Habilidades o procesos específicos dentro de cada dimensión.
- 25 preguntas: Cada una asociada a una capacidad para evaluar su implementación.
- 125 opciones de respuesta: Se ofrecen 5 opciones por pregunta, estructuradas en niveles de madurez.
- 125 planes de acción: Cada opción de respuesta tiene un plan de acción asociado para mejorar la capacidad evaluada.

Dimensiones Evaluadas

El modelo agrupa las capacidades en cinco dimensiones clave:

- Entrega Continua: Evalúa la capacidad de la organización para integrar cambios de manera rápida y confiable.
- Arquitectura: Mide la flexibilidad y escalabilidad de los sistemas.
- Producto y Procesos: Enfocado en la gestión ágil y el desarrollo basado en Lean.
- Gestión Lean y Monitoreo: Analiza cómo la empresa supervisa y optimiza sus procesos.
- Cultura: Evalúa aspectos organizacionales como la colaboración y la mejora continua.

Metodología de Evaluación

El modelo de capacidades utiliza un cuestionario estructurado, en el cual cada pregunta evalúa una capacidad específica y permite clasificar la madurez de la organización en cinco niveles:

- Sin experiencia: No se implementan prácticas DevOps.
- Novato: Uso inicial con procesos manuales.
- Intermedio: Procesos estandarizados con automatización parcial.
- Destacado: Uso avanzado de herramientas y mejores prácticas.
- Avanzado: Implementación completa y optimizada.

Cada pregunta evita la terminología técnica compleja, lo que permite aplicarla en entornos donde DevOps no está completamente adoptado.

Implementación y Análisis

Se utiliza una plataforma de encuestas para la recolección de respuestas. Los resultados se procesan en herramientas analíticas como Power BI, donde se generan reportes visuales. Se calcula un puntaje promedio para cada capacidad y dimensión, clasificándolas en:

- Oportunidades de mejora (baja madurez).
- Aceptable (madurez media).
- Fortalezas (madurez alta).

2.1.2.2. DevOps en *Hello Insight*

Hello Insight ha estado a la vanguardia en los últimos años al proveer a organizaciones sin ánimo de lucro en los Estados Unidos una herramienta accesible, capaz de evaluar el aprendizaje socio-emocional en programas extracurriculares. Esta solución permite comprender el impacto generado por dichos programas y, al mismo tiempo, ofrece recomendaciones a las entidades para potenciar ese impacto positivo en los jóvenes que atienden.

Desde sus inicios, el desarrollo de la plataforma se ha basado en principios de metodologías ágiles, integrando prácticas vinculadas al enfoque DevOps, especialmente en lo relacionado con la automatización del ciclo de vida del desarrollo. La incorporación de estas prácticas y herramientas ha sido un factor determinante en la evolución y escalabilidad del producto que *Hello Insight* ofrece.

Prácticas DevOps en *Hello Insight* *Hello Insight* cuenta con las siguientes herramientas y procesos que reflejan una adopción parcial del enfoque DevOps:

1. **Orquestación:** La orquestación está implementada mediante el uso de **Convoy**¹ sobre AWS. Esta herramienta permite gestionar despliegues automatizados, balanceo de carga y provisión de servicios en la nube. Al abstraer parte de la complejidad de la infraestructura, facilita que el equipo de desarrollo se enfoque en la evolución del producto. Asimismo, se utiliza **Docker** para la contenerización de servicios, garantizando la portabilidad entre entornos.
2. **Integración y Entrega Continua (CI/CD):** Las prácticas de CI/CD se implementaron a través de herramientas como CircleCI y GitHub Actions para ejecutar pruebas básicas y tareas de construcción (*build*) en cada envío de cambios(*push*) a ramas activas. Además, se configuraron despliegues automáticos al fusionar cambios en dichas ramas, con excepción del entorno de producción. Sin embargo, muchos de estos procesos actualmente no funcionan correctamente debido a cambios recientes en la configuración del orquestador Convoy, los cuales afectaron la integración y rompieron parte del flujo automatizado.
3. **Monitoreo:** El monitoreo se basa en herramientas básicas como **UptimeRobot**, utilizada para verificar la disponibilidad de los servicios en producción, y **Papertrail**, empleada para la centralización de logs y el análisis de errores operativos.
4. **Modularidad y arquitectura:** La arquitectura del sistema se basa en una separación clara entre **frontend**, **backend** y un módulo adicional de **analytics**, lo cual permite una gestión más eficiente, escalabilidad, y la posibilidad de desplegar componentes de forma independiente.
5. **Cultura y Procesos:** El equipo de producto fomenta la colaboración interdisciplinaria y se promueve la responsabilidad compartida.

¹Convoy. Disponible en <https://www.convoy.com/>

Desafíos de *Hello Insight* en la adopción de DevOps A pesar del uso de herramientas que evidencian una intención clara de adoptar prácticas DevOps, *Hello Insight* enfrenta varios desafíos que limitan su consolidación. Desde el punto de vista técnico, existen deficiencias en la automatización que impactan negativamente en la operación. Además, se evidencian problemas como la obsolescencia tecnológica, la falta de mantenimiento de librerías y la ausencia de procesos de prueba automatizados confiables.

En términos de entrega continua, los flujos de trabajo (*pipelines*) de despliegue han perdido estabilidad, lo cual impide una liberación constante y controlada de nuevas versiones. A nivel organizacional, no se cuenta con una cultura DevOps estructurada ni con roles definidos que promuevan la colaboración entre desarrollo, operaciones y pruebas. Esta situación se ve agravada por la falta de documentación y la carga múltiple de funciones en un equipo pequeño, lo que impide una gestión eficiente de procesos.

Asimismo, no se dispone de herramientas avanzadas de monitoreo y observabilidad, como **Prometheus**, **Grafana**, o **New Relic**, que permitan medir el rendimiento del sistema, definir alertas y anticipar fallos.

En conjunto, estas limitaciones afectan la capacidad de la organización para sostener el crecimiento del producto y ponen en riesgo la continuidad tecnológica de la plataforma.

Diagnóstico y Planificación

Con el fin de abordar el problema identificado en la adopción de prácticas DevOps en *Hello Insight*, se consideró necesario realizar un diagnóstico inicial que permitiera establecer el estado actual de la organización en relación con sus procesos, prácticas y capacidades asociadas a este enfoque, determinando así su nivel de madurez. Para ello, se empleó el modelo de capacidades de [Molina \(2022\)](#), definido en la sección 2.1.2.1, el cual no solo proporciona una valoración del nivel de madurez, sino que también entrega un plan de acción vinculado directamente con los resultados de la evaluación de capacidades. Este diagnóstico constituyó un punto de partida para comprender las fortalezas y debilidades del área de producto, así como para trazar la ruta hacia una adopción más sólida y sostenible de DevOps.

A partir de la aplicación del modelo, se obtuvo un plan de acción específico para cada capacidad evaluada. Dichos resultados fueron revisados por el Jefe de Producto y, junto con el juicio experto del equipo de desarrollo y la alineación con los objetivos empresariales, se priorizaron las capacidades que servirían como base para la elaboración de un plan de acción consolidado a ejecutar durante el mes de intervención. Como complemento a este análisis, se recolectaron los datos históricos de las métricas DORA correspondientes al 2025 (enero–agosto), lo que permitió obtener una visión más profunda y cuantitativa del estado actual de la adopción de DevOps en la organización. Este plan de acción consolidado se estructuró en actividades concretas, con un alcance temporal definido, y se diseñó con el propósito de generar impactos medibles en la mejora continua de los procesos de desarrollo y entrega de software.

En este capítulo se presentan, en primer lugar, los métodos utilizados para el diagnóstico; posteriormente, los resultados del análisis del nivel de madurez en *Hello Insight*, junto con las métricas DORA recolectadas; y, finalmente, la planificación de las acciones propuestas como respuesta a las brechas identificadas.

3.1. Modelo de Capacidades

El diagnóstico realizado en este trabajo se basó en el modelo de capacidades propuesto por [Molina \(2022\)](#). Dicho modelo ofrece una estructura sistemática para evaluar el nivel de madurez DevOps de una organización, permitiendo no solo identificar el estado actual de sus capacidades, sino también definir planes de acción de buenas prácticas DevOps. A continuación, se profundiza más en la estructura del modelo para entender su aplicación y cómo las preguntas del cuestionario están relacionadas con el plan de acción.

3.1.1. Estructura del Modelo de Capacidades de Molina (2022)

El modelo está organizado en dimensiones y capacidades específicas que abarcan aspectos de cultura, procesos, automatización y métricas. Cada capacidad se compone de un conjunto de criterios de evaluación que permiten ubicar a la organización en un nivel de madurez particular. Dichos criterios se valoran mediante preguntas, cuyas respuestas, según el nivel alcanzado, generan un plan de acción alineado con el estado diagnosticado. De esta manera, el modelo ofrece una ruta de mejora progresiva que facilita el avance desde estados iniciales hasta prácticas más consolidadas de DevOps.

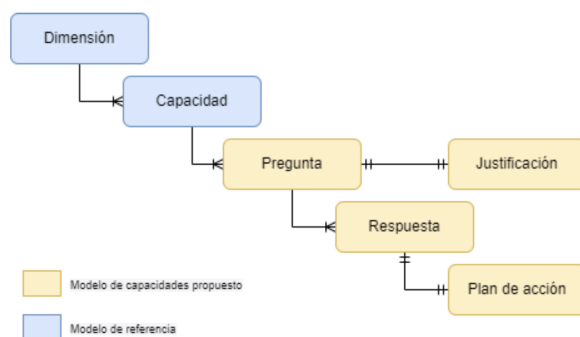


Figura 3.1: Estructura del Modelo de Capacidades propuesto por Molina. Imagen tomada de Molina (2022).

Molina define la figura 3.1 como la representación de la relación de cardinalidad entre las preguntas del modelo de capacidades y el marco de referencia. Este último organiza las capacidades dentro de categorías dimensionales, mientras que el modelo propuesto establece la correspondencia entre cada capacidad y las preguntas formuladas para evaluarla. Asimismo, cada pregunta del modelo se acompaña de una justificación que detalla el enfoque y propósito específico de la evaluación, lo que refuerza el diagnóstico y los planes de acción resultantes.

3.1.2. Estructura de las Preguntas Modelo de Capacidades de Molina (2022)

La evaluación se operacionaliza a través de un cuestionario estructurado, en el cual cada capacidad está asociada a una serie de preguntas cerradas. Estas preguntas buscan recopilar información sobre la práctica actual de la organización en relación con la dimensión evaluada, facilitando un diagnóstico objetivo y comparable. A continuación, en la figura 3.2 podemos evidenciar esta estructura.

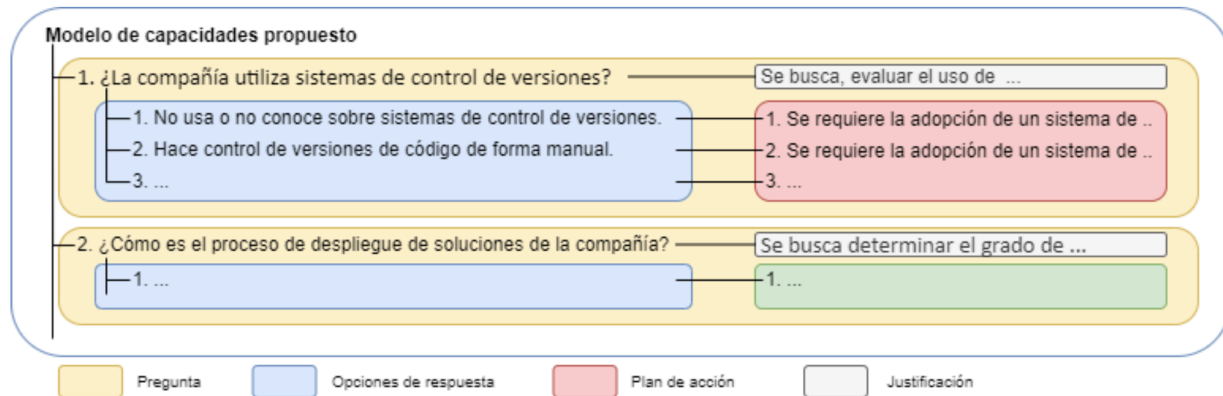


Figura 3.2: Estructura de la Pregunta del Modelo de Capacidades. Imagen tomada de [Molina \(2022\)](#).

3.1.3. Opciones de Respuesta

Cada pregunta ofrece un conjunto de opciones de respuesta que corresponden a los diferentes niveles de madurez. Estas opciones están diseñadas de forma jerárquica, de tal manera que el avance en los niveles refleja una mayor consolidación de prácticas DevOps. Estos niveles están definidos del 1 al 5, desde *Sin Experiencia* hasta *Avanzada* (Para mayor definición, los niveles se encuentran en la sección 2.1.2.1). De esta forma, las respuestas no solo sirven para determinar el nivel actual, sino que también orientan sobre las acciones necesarias para alcanzar niveles superiores.

3.1.4. Plan de Acción

El modelo no se limita a la evaluación diagnóstica, sino que incorpora un plan de acción asociado a los resultados como lo podemos ver en la figura 3.2. Este plan de acción consiste en recomendaciones específicas para cada capacidad, organizadas según el nivel alcanzado por la organización. De esta manera, el modelo proporciona una guía práctica y contextualizada para que las organizaciones puedan priorizar y planificar su camino hacia una adopción más madura de DevOps.

3.2. Implementación Encuesta de Modelo de Capacidades

Utilizando como base el proceso para usar el instrumento de evaluación definido por [Molina](#) en el punto 3.2.5, se llevaron a cabo las siguientes etapas:

3.2.1. Preparación

3.2.1.1. Reunión de Comunicación

Como primer paso, se realizó una reunión de comunicación con los integrantes del área de producto de *Hello Insight*. El objetivo de esta reunión fue socializar el propósito del diagnóstico, explicar la importancia del modelo de capacidades de Molina y resolver dudas iniciales sobre el alcance de la evaluación. De esta manera, se buscó generar un ambiente de confianza y asegurar el compromiso de los participantes en el diligenciamiento de la encuesta, enfatizando que los resultados no serían una evaluación individual, sino un insumo colectivo para mejorar los procesos de desarrollo y entrega de software.

3.2.1.2. Encuestados

Los siguientes miembros del equipo de *Hello Insight* fueron encuestados en atención a su rol dentro del área de producto, con el propósito de obtener una visión integral de los procesos y recopilar percepciones desde las responsabilidades específicas que cada uno desempeña en el ciclo de vida del producto:

- Jefe de Producto (CPO)
- Gerente de Proyecto (PM)
- Diseñador
- Líder Técnico
- Desarrollador Backend Sr.
- Desarrollador Frontend Sr.
- Desarrollador Frontend Jr.
- Director de Investigación (como reemplazo de la Analista de Producto debido a una ausencia justificada de varios meses, por lo que no fue posible obtener su participación en la evaluación previa y posterior al plan de acción)

3.2.1.3. Plataforma

Dado que la Pontificia Universidad Javeriana provee a sus estudiantes acceso a la suite de herramientas de Office 365, esta fue la plataforma seleccionada para la implementación de la encuesta. A partir de ella se extrajo la información del anexo correspondiente de Molina (2022), se realizó el montaje del cuestionario en Microsoft Forms y, gracias a la integración de la plataforma, se facilitó la recolección y consolidación de los resultados.

3.2.2. Ejecución

3.2.2.1. Construcción Cuestionario

Se llevó a cabo la preparación técnica de los instrumentos de recolección de información. En una primera etapa, las preguntas del modelo de capacidades fueron organizadas en un archivo de Excel, lo que permitió estructurar de manera sistemática las dimensiones, capacidades, opciones de respuesta y planes de acción. Posteriormente, dado que *Hello Insight* tiene como sede principal los Estados Unidos y varios de los miembros del equipo solo hablan inglés, se elaboró una traducción no oficial del archivo de Excel, en la cual se incluyeron las preguntas y respuestas en este idioma, junto con los demás datos necesarios para la correcta aplicación de la encuesta.

Finalmente, el cuestionario fue montado en Microsoft Forms, habilitando la visualización en múltiples lenguajes, lo que facilitó el acceso de los participantes. Asimismo, se asoció a un archivo de Microsoft Excel, permitiendo la recolección centralizada de las respuestas.

3.2.2.2. Apertura y cierre de la encuesta

La encuesta fue habilitada en la plataforma Microsoft Forms el 1 de agosto de 2025, en una fecha de apertura previamente comunicada al equipo de producto de *Hello Insight*, con el fin de garantizar la disponibilidad y participación de todos los miembros convocados. Durante este tiempo, se realizaron recordatorios periódicos para fomentar la participación y resolver dudas relacionadas con el diligenciamiento del cuestionario.

El cierre de la encuesta se llevó a cabo el 7 de agosto de 2025, una vez que el último participante completó sus respuestas, asegurando que todos tuvieran un tiempo suficiente para diligenciarla. Tras la recolección de la última respuesta se consolidó la información recolectada y para luego proceder con su análisis. En la tabla 3.1 se presentan las métricas obtenidas, donde se evidencian las 8 respuestas registradas, con un tiempo promedio de 38 minutos y una duración total de 7 días durante los cuales la encuesta estuvo disponible.

Apertura	Cierre	Respuestas	Tiempo Promedio	Duración
2025-08-01 08:00:00	2025-08-07 20:00:00	8	00:38	7 días

Tabla 3.1: Métricas recolectadas de la encuesta del modelo de capacidades

3.2.3. Resultados

3.2.3.1. Transformación de resultados en unidades de análisis

Para este paso se dispuso de dos archivos obtenidos en las etapas previas. Por un lado, el consolidado de preguntas, respuestas, planes de acción y demás información, al cual se denominó *Base de Datos Encuesta* (*bd_encuestas.csv*), descargado en formato CSV. Por otro lado, el archivo con las respuestas contestadas por los participantes de la encuesta, denominado *Respuestas* (*respuestas.csv*).

Utilizando Python y, específicamente, la librería `pandas`, se implementó un script para realizar la limpieza y el *scoring* de los resultados de la encuesta (véase el anexo D *Script de Procesamiento de Respuestas*).

Id	¿La compañía utiliza sistemas de control de versiones? _RID	¿Cómo es el proceso de despliegue de soluciones de la compañía? _RID	¿De qué manera realiza la compañía la compilación, construcción y empaquetamiento del producto final? _RID	¿La compañía usa alguna estrategia de gestión de la configuración? _RID	¿La compañía implementa procesos de pruebas en los desarrollos? _RID	¿Cómo definen qué datos utilizar a la hora de probar las aplicaciones? _RID	¿Cuál es el nivel de relevancia que tiene la seguridad en el ciclo de vida del desarrollo de software? _RID	¿Ejecutan tareas automáticas al registrar algún cambio? _RID	¿Qué características tiene la infraestructura de la organización? _RID	...	
0	1.0	4.0	1.0	5.0	5.0	4.0	2.0	3.0	5.0	5.0	...
1	2.0	4.0	3.0	5.0	5.0	4.0	2.0	3.0	5.0	5.0	...
2	3.0	None	1.0	5.0	3.0	None	None	None	None	None	...
3	4.0	None	3.0	None	5.0	None	None	None	2.0	None	...
4	5.0	5.0	5.0	3.0	5.0	5.0	2.0	3.0	5.0	5.0	...
5	7.0	5.0	2.0	4.0	5.0	3.0	2.0	None	4.0	1.0	...
6	8.0	4.0	2.0	5.0	5.0	5.0	None	4.0	4.0	3.0	...

Figura 3.3: Limpieza de las respuestas. Imagen propia, tomada de la herramienta Colab.

En la captura de pantalla (figura 3.3) se aprecia el paso intermedio de limpieza de las respuestas.

Previo a la última etapa del procesamiento, se decidió eliminar las opciones de respuesta *No aplica* y *No sabe/No responde* de cada pregunta, con el fin de omitir del análisis aquellas alternativas que no están alineadas con el modelo de capacidades y centrarse únicamente en los elementos definidos en el modelo propuesto.

Finalmente, se determinó que la estrategia de análisis consistiría en seleccionar el valor de menor denominación entre las respuestas de cada ítem. Por ejemplo, si en la pregunta 1 las respuestas varían entre 4, 5 y 1, el valor seleccionado es 1. Esta decisión metodológica se adoptó con la intención de identificar y atacar los problemas de raíz en las diferentes capacidades evaluadas. En este sentido, si un miembro del equipo desconoce en gran medida el funcionamiento o la existencia de una práctica o tecnología, ello constituye un indicio relevante de que la brecha puede ser mayor de lo percibido por el resto del equipo. Esta estrategia se implementó directamente en el script de Python, cuyo resultado se presenta en 3.3.

Con el fin de interpretar los resultados obtenidos en la encuesta, [Molina \(2022\)](#) define en su artefacto de software tres categorías de referencia (*benchmark*) que permiten clasificar las capacidades evaluadas según el valor alcanzado en las respuestas. Esta clasificación, como se muestra en la tabla 3.2, facilita la priorización de acciones y el diseño de estrategias de mejora continua en la adopción de prácticas DevOps.

Categoría	Rango de notas	Descripción
Oportunidades de mejora	$< 3,0$	Representa los elementos con nota baja, caracterizados como oportunidades de mejora. Son los primeros que deben ser abordados y tratados por la organización al iniciar el camino de adopción de prácticas DevOps.
Aceptable	$\geq 3,0$ y $< 4,0$	Incluye los elementos en un estado intermedio, por encima de las oportunidades de mejora pero sin alcanzar el nivel de fortaleza. Estos deben ser analizados con el fin de reforzarlos y lograr su evolución hacia el siguiente nivel.
Fortalezas	$\geq 4,0$	Define los elementos que ya se encuentran consolidados en la organización. Aunque muestran solidez, no deben ser descuidados; por el contrario, requieren ser gestionados como parte de un proceso de mejora continua.

Tabla 3.2: Clasificación de resultados según el benchmark del modelo de capacidades

En la siguiente sección se presenta el reporte consolidado del diagnóstico junto a los resultados y su correspondiente clasificación.

3.2.3.2. Reporte

A continuación se presentan los resultados con el número de la pregunta, la capacidad evaluada, los resultados obtenidos del script de Python y su correspondiente clasificación dentro de los puntos de referencia establecidos:

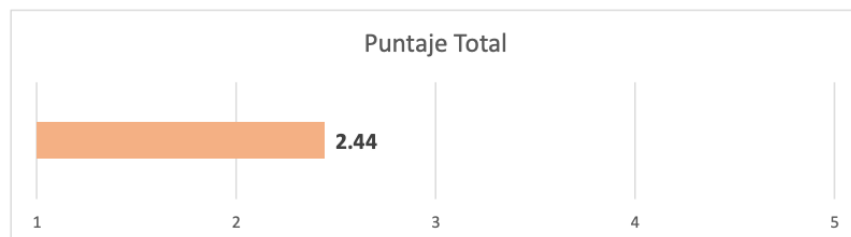
ID Pregunta	Capacidad	Resultado	Calificación
Q1	Utilizar control de versiones para todos los artefactos de producción	4	Fortalezas
Q2	Aumentar el proceso de despliegue	1	Oportunidades de mejora
Q3	Implementar integración continua	3	Aceptable
Q4	Utilizar métodos de desarrollo "Trunk-Based"	3	Aceptable
Q5	Implementar pruebas automáticas	3	Aceptable
Q6	Implementar gestión de los datos de prueba	2	Oportunidades de mejora
Q7	Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)	3	Aceptable

ID Pregunta	Capacidad	Resultado	Calificación
Q8	Implementar entrega continua de software	2	Oportunidades de mejora
Q9	Utilizar arquitecturas de bajo acoplamiento	1	Oportunidades de mejora
Q10	Permitir a los equipos definir sus propias arquitecturas	4	Fortalezas
Q11	Recolectar e implementar las recomendaciones (feedback) de los clientes	2	Oportunidades de mejora
Q12	Hacer visible el flujo de trabajo durante todo el ciclo	2	Oportunidades de mejora
Q13	Trabajar con lotes o cantidades pequeñas	3	Aceptable
Q14	Fomentar y habilitar al equipo para realizar experimentos	3	Aceptable
Q15	Tener un proceso liviano para la aprobación de cambios	1	Oportunidades de mejora
Q16	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	2	Oportunidades de mejora
Q17	Monitorizar la salud de los servicios y aplicaciones proactivamente	3	Aceptable
Q18	Administrar y establecer límites para el trabajo en progreso (Work In-Progress)	4	Fortalezas
Q19	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	1	Oportunidades de mejora
Q20	Implementar una cultura tipo generativa	2	Oportunidades de mejora
Q21	Animar y apoyar al equipo a aprender	3	Aceptable
Q22	Apoyar y facilitar la colaboración entre equipos	2	Oportunidades de mejora
Q23	Apoyar y facilitar la colaboración entre equipos	3	Aceptable
Q24	Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos	2	Oportunidades de mejora

ID Pregunta	Capacidad	Resultado	Calificación
Q25	Apoyar el liderazgo transformacional	2	Oportunidades de mejora

Tabla 3.3: Resultados obtenidos tras el diagnóstico

A partir de la tabla 3.3, que presenta los resultados individuales de cada capacidad evaluada, se elaboró un reporte consolidado con el fin de determinar el nivel actual de madurez de *Hello Insight*. Este reporte se estructuró en dos niveles de análisis: en primer lugar, una nota general obtenida a partir del promedio de todos los resultados de las dimensiones; y en segundo lugar, una nota individual calculada para cada dimensión.

Figura 3.4: Nota general del nivel de madurez DevOps en *Hello Insight*.

La figura 3.4 presenta el puntaje global obtenido a partir de la medición realizada en el área de producto de *Hello Insight*. El valor alcanzado fue de 2.44 en una escala de 1 a 5. De acuerdo con los criterios definidos en la tabla 3.2, este resultado ubica a la organización en el nivel de **Oportunidades de Mejora**, dado que se encuentra por debajo del umbral de 3.0.

Este puntaje evidencia que, aunque la organización ha comenzado a incorporar prácticas relacionadas con DevOps, aún existen brechas importantes que impiden la consolidación de procesos más avanzados y sostenibles. No obstante, el resultado también muestra que *Hello Insight* se encuentra relativamente cerca de alcanzar el siguiente nivel de madurez, lo que representa un panorama favorable para orientar esfuerzos estratégicos. Con la ejecución de acciones concretas en áreas clave la organización tiene el potencial de evolucionar rápidamente hacia categorías superiores de madurez.

En síntesis, el valor de 2.44 no solo señala las debilidades actuales, sino que también constituye una base de referencia que permite proyectar una ruta de mejora clara. Para poder continuar con el análisis debemos observar el comportamiento individual de cada dimensión.

A continuación, en la figura 3.5 se presentan los resultados individuales de cada dimensión evaluada. Ninguna de ellas supera el valor límite de 3.0, por lo que todas se clasifican dentro de la categoría de **Oportunidades de Mejora**. Asimismo, se observa que los puntajes se encuentran muy próximos entre sí, lo que sugiere la existencia de un trabajo de base homogéneo en cada dimensión. Sin embargo, este resultado también evidencia que todas requieren de un proceso estratégico y planificado que permita fortalecer sus prácticas y, de esta manera, avanzar hacia niveles superiores de madurez DevOps.

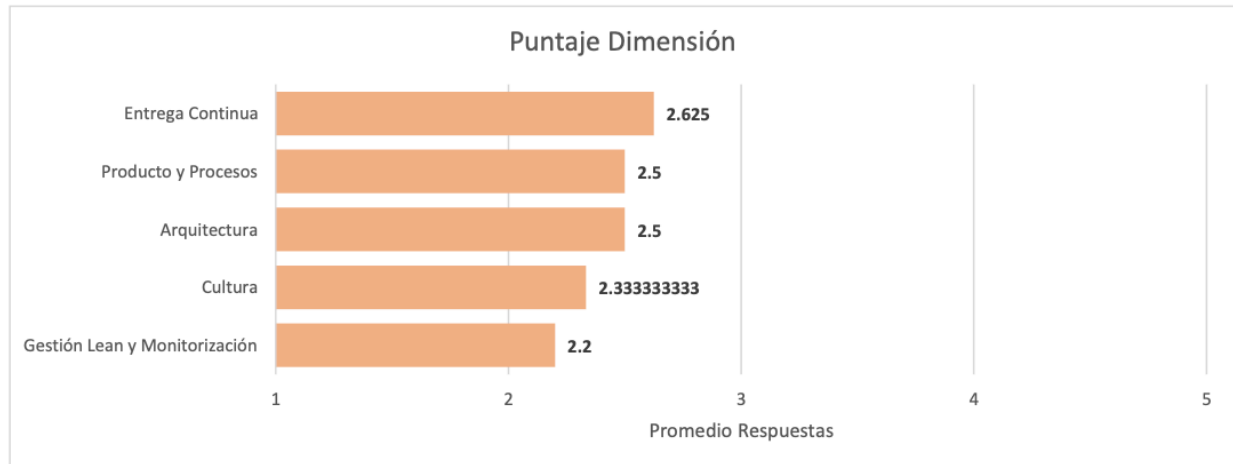


Figura 3.5: Nota por dimensión del nivel de madurez DevOps en *Hello Insight*.

Los resultados evidencian que **Entrega Continua** alcanzó el valor más alto (2.63), lo que indica que es la dimensión con mayor avance relativo dentro de la organización, aunque aún por debajo del umbral de 3.0. Le siguen **Arquitectura** y **Producto y Procesos**, ambas con un promedio de 2.50, que reflejan un estado intermedio pero con brechas importantes por atender. La dimensión de **Cultura**, con un puntaje de 2.33, confirma la necesidad de reforzar los aspectos organizacionales y de colaboración para sustentar los avances técnicos. Finalmente, **Gestión Lean y Monitorización** obtuvo la calificación más baja (2.20), lo que evidencia debilidades críticas en la gestión de cambios, monitorización y control de procesos. En conjunto, estos resultados muestran que todas las dimensiones se mantienen en el rango de **Oportunidades de Mejora**, subrayando la urgencia de un plan integral para elevar el nivel de madurez DevOps de la organización.

La figura 3.6 presenta la distribución de las capacidades evaluadas en cada dimensión, clasificadas en fortalezas, aceptables y oportunidades de mejora. Como se observa, la dimensión de **Cultura** concentra la mayor cantidad de oportunidades de mejora y no cuenta con ninguna fortaleza, lo que la posiciona como el principal reto para la organización. En contraste, **Entrega Continua** es la dimensión con más prácticas en nivel aceptable y la única que combina aceptables, fortalezas y oportunidades de mejora, reflejando avances importantes aunque aún parciales. Por su parte, **Arquitectura** y **Producto y Procesos** muestran un balance limitado entre fortalezas y oportunidades, mientras que **Gestión Lean y Monitorización** evidencia una situación intermedia, con algunas fortalezas pero también múltiples brechas. En conjunto, los resultados confirman que la organización presenta un nivel de madurez inicial, con bases distribuidas de manera uniforme pero con una alta concentración en el rango de oportunidades de mejora.



Figura 3.6: Conteo de clasificaciones por dimensión en *Hello Insight*.

La tabla 3.4 presenta un resumen final de los resultados obtenidos en cada dimensión. En ella se consolidan los promedios y la distribución porcentual de fortalezas, elementos aceptables y oportunidades de mejora, ofreciendo una visión global del nivel de madurez de cada dimensión DevOps en *Hello Insight*. El panorama general evidencia que la mayoría de las capacidades se concentran en la categoría de **Oportunidades de Mejora**, con pocas fortalezas dispersas y un grupo intermedio de prácticas en nivel aceptable.

Dimensión	Promedio	% Fort.	% Accep.	% Oport.
Arquitectura	2.50	50.00 %	0.00 %	50.00 %
Cultura	2.33	0.00 %	33.33 %	66.67 %
Entrega Continua	2.63	12.50 %	50.00 %	37.50 %
Gestión Lean y Monitorización	2.20	20.00 %	20.00 %	60.00 %
Producto y Procesos	2.50	0.00 %	50.00 %	50.00 %
Grand Total	2.44			

Tabla 3.4: Resumen: Promedio y distribución porcentual por categoría

A continuación se listan las capacidades y su dimensión según su clasificación:

Oportunidades de Mejora La tabla 3.5 presenta las capacidades identificadas con calificación baja en todas las dimensiones evaluadas. Estas capacidades representan las brechas más críticas en la adopción de prácticas DevOps, ya que su estado actual limita directamente la madurez del área de producto. Son, por tanto, los elementos que deben ser priorizados en el plan de acción inicial.

QID	Dimensión	Capacidad
Q2	Entrega Continua	Aumentar el proceso de despliegue
Q6	Entrega Continua	Implementar gestión de los datos de prueba
Q8	Entrega Continua	Implementar entrega continua de software
Q9	Arquitectura	Utilizar arquitecturas de bajo acoplamiento
Q11	Producto y Procesos	Recolectar e implementar las recomendaciones (feedback) de los clientes
Q12	Producto y Procesos	Hacer visible el flujo de trabajo durante todo el ciclo
Q15	Gestión Lean y Monitorización	Tener un proceso liviano para la aprobación de cambios
Q16	Gestión Lean y Monitorización	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio
Q19	Gestión Lean y Monitorización	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo
Q20	Cultura	Implementar una cultura tipo generativa
Q22	Cultura	Apoyar y facilitar la colaboración entre equipos
Q24	Cultura	Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos
Q25	Cultura	Apoyar el liderazgo transformacional

Tabla 3.5: Capacidades clasificadas como oportunidades de mejora

Aceptable La tabla 3.6 reúne aquellas capacidades que, si bien han alcanzado un estado intermedio, todavía no se consolidan como fortalezas. Estas representan prácticas con un nivel de madurez en desarrollo y que, con un refuerzo adecuado, pueden convertirse en fortalezas a corto o mediano plazo.

Dimensión	Capacidad
Entrega Continua	Implementar integración continua
Entrega Continua	Utilizar métodos de desarrollo “Trunk-Based”
Entrega Continua	Implementar pruebas automáticas
Entrega Continua	Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)
Producto y Procesos	Trabajar con lotes o cantidades pequeñas
Producto y Procesos	Fomentar y habilitar al equipo para realizar experimentos
Gestión Lean y Monitorización	Monitorizar la salud de los servicios y aplicaciones proactivamente
Cultura	Animar y apoyar al equipo a aprender
Cultura	Apoyar y facilitar la colaboración entre equipos

Tabla 3.6: Capacidades clasificadas como aceptables

Fortalezas La tabla 3.7 identifica las capacidades que ya se encuentran consolidadas dentro de la organización. Estas prácticas representan un soporte clave sobre el cual se puede apalancar la mejora continua. Sin embargo, no deben descuidarse, pues requieren mantenerse vigentes y alineadas con la evolución del resto de dimensiones.

Dimensión	Capacidad
Entrega Continua	Utilizar control de versiones para todos los artefactos de producción
Arquitectura	Permitir a los equipos definir sus propias arquitecturas
Gestión Lean y Monitorización	Administrar y establecer límites para el trabajo en progreso (Work In-Progress)

Tabla 3.7: Capacidades clasificadas como fortalezas

Análisis Final El análisis consolidado de los resultados evidencia que la organización se encuentra en un nivel de madurez inicial, donde la mayoría de las capacidades evaluadas se concentran en la categoría de **Oportunidades de Mejora**. Tal como se observó en las tablas de resumen, ninguna dimensión supera el umbral de 3.0, lo que confirma que todas requieren atención estratégica. Entre ellas, destacan **Cultura** y **Gestión Lean y Monitorización** como las dimensiones más críticas, dado que concentran el mayor porcentaje de oportunidades de mejora y carecen de fortalezas consolidadas. En contraste, **Entrega Continua** mostró un desempeño ligeramente superior, con el promedio más alto y la mayor proporción de capacidades en nivel aceptable, lo que la posiciona como la base más sólida para apalancar mejoras.

A pesar de que algunas dimensiones presentan un mayor número de debilidades que otras, el enfoque adoptado para este trabajo prioriza las **Oportunidades de Mejora** como eje transversal a nivel de capacidades, ya que constituyen el punto común desde el cual es posible articular intervenciones de alto impacto y definir una ruta inicial de crecimiento para todas las dimensiones del modelo. Desde una perspectiva estratégica, centrarse en estas oportunidades permite generar el mayor impacto con el menor esfuerzo, dado que estas capacidades suelen influir simultáneamente en varias dimensiones y su fortalecimiento produce beneficios inmediatos en la estabilidad, la colaboración y la eficiencia del flujo de trabajo.

De acuerdo con los resultados obtenidos al aplicar el modelo de Molina (2022), dichas capacidades se encuentran en un estado insuficiente para soportar prácticas más avanzadas, por lo que intervenirlas en primera instancia permite reducir debilidades estructurales y habilitar condiciones fundamentales para la evolución del proceso DevOps.

En síntesis, el análisis permitió identificar las capacidades clasificadas como **Oportunidades de Mejora**, junto con los planes de acción preliminares generados a partir del artefacto de Molina. Estas capacidades constituyen los puntos con mayor necesidad de intervención y la base para estructurar el plan de acción definitivo adaptado al contexto organizacional, las particularidades del equipo y las necesidades específicas de *Hello Insight*.

El detalle completo de las capacidades con el plan de acción sugerido por el artefacto se presenta en el anexo A, el cual puede consultarse para ampliar la información y revisar la salida exacta del

instrumento.

3.3. Métricas DORA 2025 (Enero - Agosto) en Hello Insight

En esta sección se presentan los resultados de las métricas DORA aplicadas a los tres servicios principales de *Hello Insight*: HI-API, HI-Frontend y HI-Analytics. A su vez, este apartado complementa el diagnóstico realizado mediante el modelo de capacidades, al incorporar una lectura cuantitativa del desempeño de entrega durante el periodo de 2025. Cada servicio se analiza por separado con el fin de ofrecer una visión más precisa de su comportamiento en términos de velocidad y estabilidad en la entrega de software. En términos generales, los tres servicios se ubican en el nivel de **Desempeño Medio (Medium Performer)** en la mayoría de las métricas evaluadas.

Las métricas fueron recolectadas a partir de diversas fuentes operativas internas: (i) los registros de despliegue generados por la plataforma *Convoox*, (ii) los commits capturados desde los repositorios asociados, y (iii) los incidentes y errores reportados mediante el sistema de monitoreo *Sentry*. Este conjunto de datos permitió caracterizar el comportamiento real de los servicios durante el periodo comprendido entre enero y agosto de 2025.

Para la interpretación de los resultados se utilizaron los umbrales de referencia establecidos por [DevOps Research and Assessment \(DORA\) \(2023\)](#), complementados con los criterios de clasificación definidos en [Apache Software Foundation \(2023\)](#). La combinación de ambos marcos permitió evaluar cada repositorio bajo parámetros estandarizados y comparables, alineados con las prácticas internacionales de medición de desempeño DevOps.

3.3.1. HI-API

El servicio HI-API representa la capa de backend y expone la lógica principal del producto. A continuación se presentan los resultados correspondientes a las métricas DORA.

3.3.1.1. Frecuencia de Despliegue

Grupo	Referencia	Criterio DevLake	HI-API (2025)
Élite	Bajo demanda (múltiples despliegues por día)	5 días/semana	
Alto desempeño	1 día – 1 semana	1 día/semana	
Desempeño medio	1 semana – 1 mes	1 día/mes	3.5 despliegues/mes
Bajo desempeño	<1 día/mes	<1 día/mes	

Tabla 3.8: Frecuencia de despliegue HI-API (2025)

3.3.1.2. Tiempo de Recuperación de Despliegue Fallido (MTTR)

Grupo	Referencia	Benchmark DevLake	HI-API
Élite	<1 hora	<1 hora	
Alto desempeño	<1 día	<1 día	
Desempeño medio	1 día – 1 semana	1 día – 1 semana	2.5 días
Bajo desempeño	>6 meses	>1 semana	

Tabla 3.9: Tiempo de recuperación HI-API

3.3.1.3. Tiempo de Entrega de Cambios (Lead Time for Changes)

Grupo	Referencia	Benchmark DevLake	HI-API
Élite	<1 día	<1 día	
Alto desempeño	1 día – 1 semana	1 día – 1 semana	
Desempeño medio	1 semana – 1 mes	1 semana – 1 mes	X
Bajo desempeño	>1 mes	>1 mes	

Tabla 3.10: Lead Time for Changes HI-API

3.3.1.4. Tasa de Fallos en Cambios (Change Failure Rate)

Grupo	Referencia	Benchmark DevLake	HI-API
Élite	5 %	(0, 5 %]	
Alto desempeño	10 %	(5 %, 10 %]	
Desempeño medio	15 %	(10 %, 15 %]	13.79 %
Bajo desempeño	64 %	(15 %, 100 %]	

Tabla 3.11: Change Failure Rate HI-API

Resumen HI-API

En el caso de *HI-API*, los resultados de las métricas DORA para el periodo enero–agosto de 2025 muestran un comportamiento consistente en el nivel de **Desempeño Medio**. La frecuencia de despliegue se ubica en la categoría de desempeño medio, con una mediana de aproximadamente **3.5 despliegues al mes**, lo que indica una cadencia regular pero aún distante de los niveles de despliegue semanal o bajo demanda.

El *Lead Time for Changes* se clasifica también en el rango de **1 semana a 1 mes**, alineado con el nivel de **Desempeño Medio**, lo que sugiere que los cambios tardan varios días en pasar de

commit a producción. De forma complementaria, el *Mean Time to Recover* (MTTR) se ubica en el mismo rango de desempeño medio, con un tiempo de recuperación aproximado de **2.5 días** ante despliegues fallidos.

Finalmente, la *Change Failure Rate* para HI-API es de **13.79%**, valor que se encuentra dentro del intervalo de **Desempeño Medio** (10 %–15 %). En conjunto, estos resultados reflejan un servicio con una base estable de prácticas DevOps, pero con oportunidades claras para reducir tiempos de entrega y de recuperación, así como para disminuir la proporción de cambios fallidos.

3.3.2. HI-Frontend

El componente de frontend corresponde a la interfaz principal utilizada por los clientes. A continuación se muestran las métricas asociadas.

3.3.2.1. Frecuencia de Despliegue

Grupo	Referencia	Criterio DevLake	HI-FRONTEND (2025)
Élite	Bajo demanda	5 días/semana	
Alto desempeño	1 día – 1 semana	1 día/semana	1 despliegue/semana
Desempeño medio	1 semana – 1 mes	1 día/mes	
Bajo desempeño	<1 día/mes	<1 día/mes	

Tabla 3.12: Frecuencia de despliegue HI-FRONTEND (2025)

3.3.2.2. Tiempo de Recuperación de Despliegue Fallido (MTTR)

Grupo	Referencia	Benchmark DevLake	HI-FRONTEND
Élite	<1 hora	<1 hora	
Alto desempeño	<1 día	<1 día	
Desempeño medio	1 día – 1 semana	1 día – 1 semana	3 días
Bajo desempeño	>6 meses	>1 semana	

Tabla 3.13: Tiempo de recuperación HI-FRONTEND

3.3.2.3. Tiempo de Entrega de Cambios (Lead Time for Changes)

Grupo	Referencia	Benchmark DevLake	HI-FRONTEND
Élite	<1 día	<1 día	
Alto desempeño	1 día – 1 semana	1 día – 1 semana	
Desempeño medio	1 semana – 1 mes	1 semana – 1 mes	X
Bajo desempeño	>1 mes	>1 mes	

Tabla 3.14: Lead Time for Changes HI-FRONTEND

3.3.2.4. Tasa de Fallos en Cambios (Change Failure Rate)

Grupo	Referencia	Benchmark DevLake	HI-FRONTEND
Élite	5 %	(0, 5 %]	
Alto desempeño	10 %	(5 %, 10 %]	
Desempeño medio	15 %	(10 %, 15 %]	13.04 %
Bajo desempeño	64 %	(15 %, 100 %]	

Tabla 3.15: Change Failure Rate HI-FRONTEND

Resumen HI-Frontend

Para *HI-Frontend*, las métricas DORA evidencian un comportamiento ligeramente más avanzado en términos de velocidad de entrega. La frecuencia de despliegue en 2025 alcanza una mediana de **un despliegue por semana**, lo que sitúa a este servicio en la categoría de **Alto desempeño** para la métrica de *Deployment Frequency*. Este resultado indica una mayor agilidad para llevar cambios a producción en comparación con los demás servicios.

No obstante, el resto de métricas se mantienen en el rango de **Desempeño Medio**. El *Lead Time for Changes* se encuentra entre **una semana y un mes**, y el *Mean Time to Recover* (MTTR) se aproxima a **3 días** ante despliegues fallidos, lo que implica que, aunque se despliega con frecuencia, la recuperación ante errores aún requiere varios días. Por su parte, la *Change Failure Rate* es de **13.04 %**, también clasificada como desempeño medio.

En conjunto, HI-Frontend presenta una **buena cadencia de despliegue** con resultados intermedios en estabilidad, lo que sugiere que los esfuerzos futuros deberían centrarse en reducir el tiempo de entrega y de recuperación, así como en mejorar la calidad de los cambios para disminuir la tasa de fallos.

3.3.3. HI-Analytics

Este servicio está asociado a la capa analítica y al procesamiento de datos. A continuación, se presentan los resultados correspondientes.

3.3.3.1. Frecuencia de Despliegue

Grupo	Referencia	Criterio DevLake	HI-ANALYTICS (2025)
Élite	Bajo demanda	5 días/semana	
Alto desempeño	1 día – 1 semana	1 día/semana	
Desempeño medio	1 semana – 1 mes	1 día/mes	1 despliegue/mes
Bajo desempeño	<1 día/mes	<1 día/mes	

Tabla 3.16: Frecuencia de despliegue HI-ANALYTICS (2025)

3.3.3.2. Tiempo de Recuperación de Despliegue Fallido (MTTR)

Grupo	Referencia	Benchmark DevLake	HI-ANALYTICS
Élite	<1 hora	<1 hora	
Alto desempeño	<1 día	<1 día	
Desempeño medio	1 día – 1 semana	1 día – 1 semana	
Bajo desempeño	>6 meses	>1 semana	19 días

Tabla 3.17: Tiempo de recuperación HI-ANALYTICS

3.3.3.3. Tiempo de Entrega de Cambios (Lead Time for Changes)

Grupo	Referencia	Benchmark DevLake	HI-ANALYTICS
Élite	<1 día	<1 día	
Alto desempeño	1 día – 1 semana	1 día – 1 semana	
Desempeño medio	1 semana – 1 mes	1 semana – 1 mes	X
Bajo desempeño	>1 mes	>1 mes	

Tabla 3.18: Lead Time for Changes HI-ANALYTICS

3.3.3.4. Tasa de Fallos en Cambios (Change Failure Rate)

Grupo	Referencia	Benchmark DevLake	HI-ANALYTICS
Élite	5 %	(0, 5 %]	
Alto desempeño	10 %	(5 %, 10 %]	
Desempeño medio	15 %	(10 %, 15 %]	11.11 %
Bajo desempeño	64 %	(15 %, 100 %]	

Tabla 3.19: Change Failure Rate HI-ANALYTICS

Resumen HI-Analytics

En el caso de *HI-Analytics*, los resultados muestran un comportamiento más heterogéneo. La frecuencia de despliegue en 2025 se sitúa en **un despliegue por mes**, lo que ubica a este servicio en el nivel de **Desempeño Medio** para *Deployment Frequency*. De igual forma, el *Lead Time for Changes* se encuentra en el rango de **una semana a un mes**, coherente con un nivel intermedio de madurez en la entrega de cambios.

Sin embargo, la métrica de *Mean Time to Recover* (MTTR) revela una debilidad importante: el tiempo medio de recuperación ante un despliegue fallido es de aproximadamente **19 días**, un valor que excede el umbral de una semana y se clasifica como **Bajo desempeño**. Esto indica que, cuando se presentan incidentes en HI-Analytics, el restablecimiento del servicio puede tomar varias semanas, afectando la estabilidad operativa.

Por otro lado, la *Change Failure Rate* es del **11.11 %**, lo que se ubica en el rango de **Desempeño Medio**. En síntesis, HI-Analytics muestra una adopción intermedia de prácticas DevOps en cuanto a frecuencia de despliegue, tiempos de entrega y calidad de cambios, pero evidencia una **brecha crítica en el tiempo de recuperación**, que debe ser abordada en futuros ciclos de mejora para fortalecer su confiabilidad.

3.4. Socialización de Resultados y Priorización de Capacidades

El proceso que condujo a la priorización de capacidades se desarrolló mediante una secuencia de actividades encadenadas que inició con la recolección de información y culminó en un consenso entre el Jefe de Producto y el equipo de desarrollo. En primer lugar, a partir de los datos obtenidos mediante la aplicación del modelo de capacidades y el análisis de las métricas DORA, se elaboró un reporte consolidado que incluía los hallazgos principales, las métricas recolectadas y las oportunidades de mejora identificadas.

Este reporte fue socializado inicialmente con el Jefe de Producto (*Chief Product Officer*) y con los demás integrantes del área de producto durante una reunión diseñada para presentar los resultados de forma gráfica y textual. Este espacio permitió aclarar dudas, explicar el alcance de los hallazgos y generar un entendimiento compartido sobre el estado de madurez de *Hello Insight*.

La retroalimentación obtenida en esta sesión resultó fundamental para alinear la interpretación de los resultados con los objetivos estratégicos de la organización y para fortalecer el compromiso del equipo frente a las acciones propuestas.

Posteriormente, se llevó a cabo una reunión individual con el Jefe de Producto, en la cual se revisó en detalle el significado, alcance e implicaciones de cada una de las capacidades identificadas como oportunidades de mejora. La priorización se definió a partir de tres criterios, alineados con el diagnóstico previo y con la sección de métricas: (i) la brecha de madurez evidenciada por el modelo de capacidades, (ii) el impacto esperado sobre el desempeño de entrega observado mediante las métricas DORA, y (iii) la viabilidad de implementación dentro del horizonte de cuatro semanas (esfuerzo estimado, dependencias y control del equipo sobre los cambios). Con base en estos criterios, el Jefe de Producto construyó una primera versión de priorización individual y se aplicó una escala de tres niveles: **prioridad 1** para capacidades con brecha alta, impacto directo en el desempeño y alta viabilidad en el corto plazo; **prioridad 2** para capacidades con impacto y brecha relevantes, pero con dependencias que requieren una ejecución progresiva; y **prioridad 3** para capacidades que, aunque necesarias, demandan transformaciones organizacionales o culturales de mayor alcance y, por tanto, se planifican para una fase posterior.

De forma complementaria, se efectuó una última socialización con el equipo de desarrollo. En esta sesión, los miembros del equipo —desde sus distintos roles técnicos y operativos— realizaron una priorización individual de las mismas capacidades, considerando las necesidades técnicas, el esfuerzo requerido y las dependencias de implementación. Como resultado, se obtuvieron dos listas de priorización preliminar: una del Jefe de Producto y otra del equipo de desarrollo. El análisis comparado permitió identificar puntos de coincidencia y diferencias derivadas de perspectivas complementarias.

En la Tabla 3.20 se presenta la comparación entre las priorizaciones preliminares del Jefe de Producto y del equipo de desarrollo, junto con la elección final, para el conjunto completo de capacidades clasificadas. En particular, en las capacidades ubicadas como **prioridad 3** se evidenció un acuerdo entre ambas partes en asignarlas al final, debido a que requieren un mayor alcance organizacional o no resultan viables dentro del horizonte de cuatro semanas. Por el contrario, en las capacidades de **prioridad 1** y **prioridad 2** se presentaron diferencias en la priorización preliminar. Como se observa, existió consenso inicial en dos capacidades clasificadas como **prioridad 1** por ambas partes, mientras que en las restantes el equipo se inclinó principalmente por aspectos técnicos vinculados a la Entrega Continua —por ejemplo, “Aumentar el proceso de despliegue” e “Implementar entrega continua de software”—, y el Jefe de Producto priorizó capacidades asociadas con Gestión Lean y Monitorización —por ejemplo, “Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio” y “Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo”—.

Capacidad	Jefe de Producto	Equipo	Elección
Implementar gestión de los datos de prueba	1	1	1
Utilizar arquitecturas de bajo acoplamiento	1	1	1
Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	1	2	1
Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	1	2	1
Aumentar el proceso de despliegue	2	1	2
Implementar entrega continua de software	2	1	2
Tener un proceso liviano para la aprobación de cambios	2	2	2
Apoyar y facilitar la colaboración entre equipos	2	2	2
Apoyar el liderazgo transformacional	2	2	2
Hacer visible el flujo de trabajo durante todo el ciclo	3	3	3
Implementar una cultura tipo generativa	3	3	3
Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos	3	3	3
Recolectar e implementar las recomendaciones (feedback) de los clientes	3	3	3

Tabla 3.20: Comparación de priorización preliminar (Jefe de Producto vs. Equipo) y priorización final consensuada

A partir de este contraste y de la discusión conjunta, se consolidó el conjunto final de capacidades priorizadas, clasificadas en los niveles **1**, **2** y **3**. Esta priorización definitiva constituye el insumo principal para la construcción del *Plan de Acción*, ya que determina qué capacidades se abordan de forma inmediata (prioridad 1) y cuáles se dejan documentadas para una o varias fases posteriores (prioridad 2 y 3). En la Sección 3.5.1 se presentan las capacidades priorizadas consolidadas y su nivel de priorización definitiva, las cuales serán desagregadas en tareas y organizadas en el cronograma de implementación.

3.5. Plan de Acción

3.5.1. Capacidades Priorizadas

Como resultado de los pasos anteriores, se obtuvo una lista de capacidades priorizadas que sirvieron como el insumo para la elaboración del plan de acción. Estas capacidades fueron posteriormente desagregadas en tareas específicas y organizadas en un cronograma de implementación con una duración de un mes.

En la tabla 3.21 se presentan las dimensiones y capacidades identificadas junto a su nivel de priorización. El valor asignado va de 1 a 3, donde **1** corresponde a las capacidades más críticas, seleccionadas para conformar el plan de acción inmediato; mientras que los niveles **2** y **3** representan capacidades que, si bien no se atenderán en la fase inicial, constituyen la base para la planificación de mejoras a futuro.

Categoría	Capacidad	Priorización
Arquitectura	Utilizar arquitecturas de bajo acoplamiento	1
Entrega Continua	Implementar gestión de los datos de prueba	1
Gestión Lean y Monitorización	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	1
Gestión Lean y Monitorización	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	1
Cultura	Apoyar y facilitar la colaboración entre equipos	2
Cultura	Apoyar el liderazgo transformacional	2
Entrega Continua	Aumentar el proceso de despliegue	2
Entrega Continua	Implementar entrega continua de software	2
Gestión Lean y Monitorización	Tener un proceso liviano para la aprobación de cambios	2
Producto y Procesos	Hacer visible el flujo de trabajo durante todo el ciclo	3
Cultura	Implementar una cultura tipo generativa	3
Cultura	Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos	3
Producto y Procesos	Recolectar e implementar las recomendaciones (feedback) de los clientes	3

Tabla 3.21: Capacidades priorizadas para el plan de acción

Capacidades de priorización 1 Las capacidades con nivel de priorización **1** se distribuyen entre tres dimensiones clave: *Arquitectura*, *Entrega Continua* y *Gestión Lean y Monitorización*. La elección de estas capacidades como más críticas es coherente con los resultados obtenidos en las métricas de madurez, pues dos de estas dimensiones se encontraban entre las más bajas en los promedios. En particular, la inclusión de *Entrega Continua* refleja que, aunque fue la dimensión

con mayor avance relativo, aún existen vacíos fundamentales que son importantes para cumplir con los objetivos de la empresa.

Estas capacidades resultan prioritarias porque inciden directamente en la eficiencia técnica y en la capacidad de responder de forma ágil a las necesidades de los clientes.

Capacidades de priorización 2 Las capacidades clasificadas con nivel de priorización **2** abarcan un conjunto más amplio de dimensiones, incluyendo *Cultura*, *Entrega Continua* y *Gestión Lean y Monitorización*. Su ubicación en este nivel intermedio tiene sentido porque representan aspectos estructurales que, si bien no son las debilidades más urgentes, son necesarios para garantizar que las mejoras iniciales tengan continuidad. Además, la presencia de *Cultura* en este grupo muestra que el equipo reconoció la importancia de fortalecer prácticas de colaboración y liderazgo, aunque se decidió postergar su atención inmediata para priorizar primero las bases técnicas.

Estas capacidades, aunque no críticas de forma inmediata, representan el siguiente nivel de avance y consolidación de la madurez DevOps.

Capacidades de priorización 3 Las capacidades ubicadas en el nivel de priorización **3** corresponden principalmente a *Cultura* y *Producto y Procesos*. Esta decisión es coherente con el hecho de que, aunque estas dimensiones obtuvieron promedios bajos en la evaluación, las capacidades asociadas requieren transformaciones de mayor alcance organizacional y cultural que no son viables en el corto plazo. Por lo tanto, se reconocen como metas estratégicas de mediano y largo plazo, que podrán abordarse una vez se hayan cerrado las brechas críticas de carácter técnico y de gestión.

Si bien estas capacidades no requieren atención inmediata, permitirán consolidar las prácticas ya fortalecidas en las fases previas.

Síntesis general En conjunto, la priorización muestra que los esfuerzos iniciales deben enfocarse en las debilidades críticas de tipo técnico y organizacional (prioridad 1), complementarse con mejoras estructurales en procesos y cultura (prioridad 2), y finalmente proyectarse hacia una transformación organizacional más amplia y sostenible (prioridad 3). Esta hoja de ruta escalonada permite que los avances se construyan de manera progresiva y alineada con los objetivos estratégicos de la organización.

Para dar continuidad al proceso, se realizó una división de tareas específicas para cada una de las capacidades clasificadas con priorización **1**, tomando como base el plan de acción recomendado por [Molina \(2022\)](#), los resultados de la encuesta y el juicio de experto del equipo de desarrollo. Durante este ejercicio se evaluó que algunos elementos del plan de acción ya se encontraban implementados parcialmente, identificándose que la principal brecha estaba relacionada con la ausencia de una comunicación clara hacia el equipo sobre la infraestructura y los flujos de despliegue existentes.

Es importante señalar que, si bien las tareas definidas pueden tener impacto en otras capacidades, para efectos de este proyecto se pondrá el foco principalmente en las cuatro capacidades priorizadas como nivel 1, de manera que se logre una intervención concreta y medible en los aspectos más críticos para la madurez DevOps del área de producto en *Hello Insight*.

3.5.2. Actividades

Lluvia de ideas de actividades Según Wilson (2010), la lluvia de ideas es una técnica grupal orientada a generar el mayor número posible de propuestas en un entorno libre de juicios, con el fin de estimular la creatividad y el pensamiento divergente. Esta técnica reúne a un grupo diverso de participantes para aportar ideas de manera espontánea, permitiendo construir un conjunto amplio de posibles actividades que posteriormente pueden organizarse, depurarse y priorizarse. En este proyecto, la lluvia de ideas fue seleccionada como herramienta metodológica para facilitar la exploración inicial de actividades asociadas a cada capacidad priorizada, dado que favorece la participación abierta, la generación rápida de alternativas y la construcción colaborativa de insumos para el diseño del plan de acción.

En este caso, se tiene un contexto de base compuesto por los resultados de la priorización y los planes de acción obtenidos a partir del artefacto propuesto en Molina (2022). El tiempo asignado para cada capacidad fue de 15 minutos. Para la ejecución de la dinámica se empleó la estrategia de *post-its* de diferentes colores, en los cuales cada participante consignó las tareas que consideraba pertinentes para cada capacidad. Con el fin de guiar el ejercicio, se definió un tablero blanco (*whiteboard*) con ejemplos de referencia, tal como se muestra en la Figura 3.7.

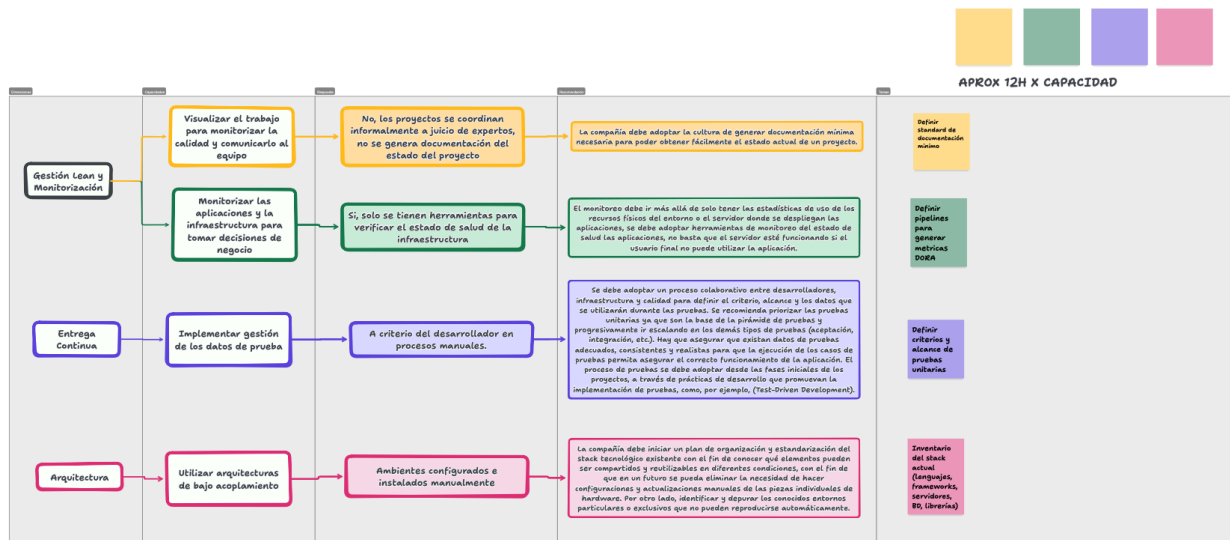


Figura 3.7: Whiteboard para lluvia de ideas.

Resultado Lluvia de ideas La participación de los integrantes en la actividad fue activa y constante. En promedio, se recolectaron seis o más propuestas de actividades por cada plan de acción de cada capacidad. Las actividades registradas serán objeto de un proceso de depuración y análisis. Estas iniciativas se someterán a una evaluación de tiempos estimados de ejecución (ETA)

con el propósito de identificar aquellas más viables para ser implementadas en el siguiente mes de trabajo y así obtener el cronograma final.

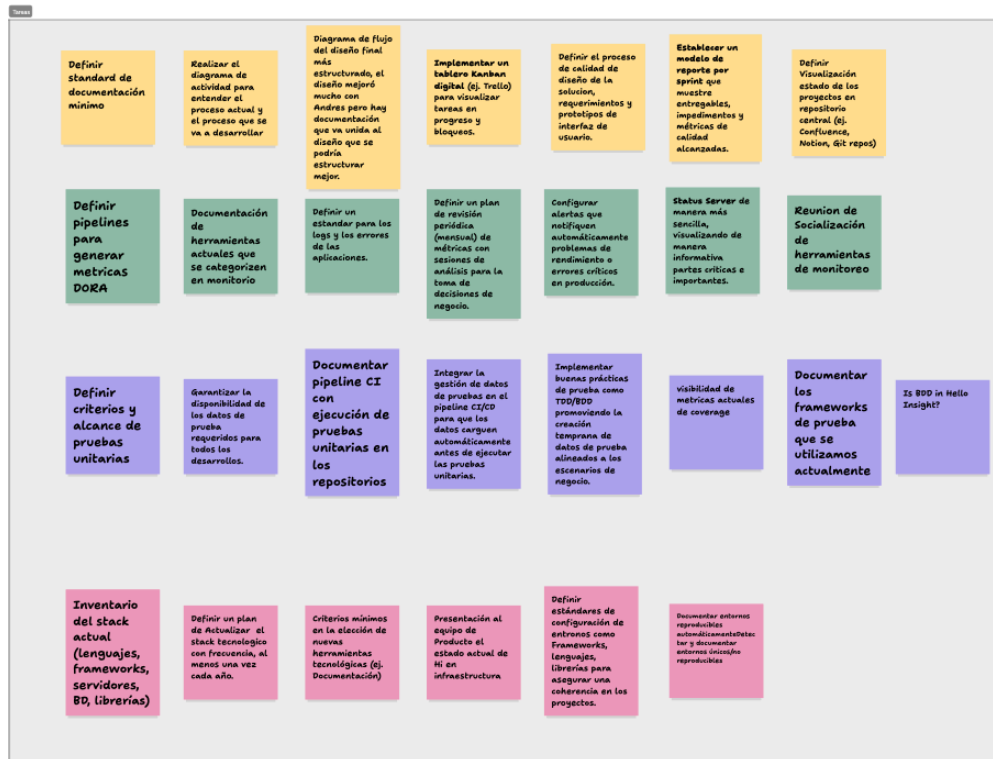


Figura 3.8: Resultado de la lluvia de ideas.

Los participantes de la actividad fueron los miembros del equipo de desarrollo —líder técnico, desarrollador Frontend Sr., desarrolladora Frontend Jr. y desarrollador Backend— junto con el Jefe de Producto, quienes aportaron perspectivas complementarias que enriquecieron la generación de propuestas.

3.5.3. Definición de Actividades

Las actividades definidas para cada capacidad surgieron de un proceso iterativo que combinó los insumos obtenidos en la lluvia de ideas realizada con el equipo de producto, junto con los planes de acción propuestos en el artefacto de [Molina Molina \(2022\)](#). A partir de estos elementos se desarrollaron líneas de trabajo adaptadas al contexto de *Hello Insight*, enfocadas en atender las brechas identificadas durante la evaluación de madurez. En esta sección se describe, para cada capacidad priorizada, el enfoque general y los propósitos estratégicos de las actividades diseñadas. El detalle completo de las actividades, incluyendo esfuerzo estimado, entregables y tiempos de ejecución, se presenta en el anexo B.

3.5.3.1. Entrega Continua — Gestión de datos de prueba

El enfoque de esta capacidad se centró en fortalecer la base del proceso de pruebas, asegurando que el equipo contara con datos confiables, accesibles y consistentes para validar el comportamiento del software en distintas etapas del ciclo de vida. Las actividades diseñadas apuntaron a documentar el estado actual de las pruebas, definir un marco de alcance para pruebas unitarias en los diferentes módulos del sistema, y establecer un proceso formal para la creación, disponibilidad y mantenimiento de los datos de prueba. Asimismo, se contemplaron acciones orientadas a integrar la ejecución de pruebas en el *pipeline* de CI, mejorar la visibilidad de la cobertura y establecer umbrales mínimos para garantizar la calidad. En conjunto, estas iniciativas buscan consolidar una base sólida para prácticas de entrega continua sostenibles.

3.5.3.2. Arquitectura — Bajo acoplamiento

El trabajo en esta capacidad se enfocó en promover una arquitectura más estandarizada, escalable y fácil de mantener. Para ello, se plantearon actividades orientadas a documentar la estructura actual mediante diagramas C4, identificar el estado del stack tecnológico y sus riesgos, y establecer criterios claros para la incorporación de nuevas herramientas en la organización. El objetivo general fue entender el estado actual, mejorar la claridad arquitectónica y habilitar prácticas que favorezcan la reproducibilidad y la toma de decisiones técnicas fundamentadas.

3.5.3.3. Gestión Lean y Monitorización — Aplicaciones e Infraestructura

El enfoque principal de esta capacidad consistió en ampliar la visibilidad operacional más allá de métricas de infraestructura, incorporando indicadores relacionados con experiencia de usuario y desempeño del negocio. Las actividades definidas buscaron consolidar y documentar las herramientas actuales de monitorización, establecer prácticas de revisión periódica, y desarrollar mecanismos automáticos para la recolección y publicación de métricas DORA. Además, se planearon acciones para implementar monitorización sintética y definir al menos un SLI/SLO alineado con un servicio crítico. Este conjunto de esfuerzos apunta a fortalecer la toma de decisiones basada en datos y a mejorar la capacidad del equipo para detectar y responder a incidentes.

3.5.3.4. Gestión Lean y Monitorización — Visualizar el trabajo

El propósito de esta capacidad fue establecer mecanismos de documentación mínima y visualización que permitieran al equipo comprender el estado de los proyectos de forma clara y oportuna. Las actividades asociadas incluyeron la definición de estándares de documentación, la creación de procesos de revisión de calidad, la construcción de tableros Kanban estandarizados y la consolidación de una visualización centralizada del estado de avance. Asimismo, se propuso un ritual quincenal de revisión para garantizar la vigencia de la documentación y mantener una cultura de transparencia operativa. Estos elementos contribuyen a mejorar la comunicación interna y a facilitar la coordinación del trabajo.

3.5.4. Resumen consolidado

La siguiente Tabla 3.22 presenta una visión global del esfuerzo estimado para las actividades definidas en cada dimensión. Aunque el detalle operativo de las tareas se encuentra en el anexo B, este resumen permite identificar la distribución del trabajo por dimensión, el volumen de actividades involucradas y la carga de esfuerzo proyectada para las cuatro semanas del plan de acción.

Dimensión	Nº de tareas	Esfuerzo total (h)	Semanas
Entrega Continua	6	14	1–4
Arquitectura	4	14	1–4
Monitorización (Apps/Infra)	7	26	1–4
Visualizar el trabajo	6	20	1–4
Total	23	74	1–4

Tabla 3.22: Resumen consolidado de esfuerzo estimado por dimensión para 4 semanas.

3.6. Cronograma de Actividades

Esta sección presenta el cronograma de ejecución del plan de acción, organizado en cuatro semanas y alineado con el resumen consolidado de actividades y esfuerzo (Tabla 3.22). El cronograma funciona como una guía operativa para secuenciar las tareas, evidenciar dependencias y asegurar una progresión ordenada desde actividades técnicas hasta actividades orientadas a la coordinación, alineación y socialización de resultados. La Figura 3.9 muestra el cronograma de actividades resultante a partir de lo definido en el resumen consolidado.

En términos generales, la planificación prioriza al inicio las tareas de mayor componente técnico —particularmente aquellas asociadas a Entrega Continua, Arquitectura y Monitorización de aplicaciones e infraestructura— con el fin de establecer bases de automatización, observabilidad y consistencia técnica que habiliten el resto del trabajo. A medida que avanza el plan, se incrementa la atención sobre actividades de visualización del trabajo, estandarización de prácticas y coordinación del equipo. Finalmente, durante la última etapa se concentran las actividades de validación, cierre y socialización, orientadas a compartir hallazgos, consolidar aprendizajes y formalizar acuerdos para la sostenibilidad de las mejoras implementadas.

Actividades	Semanas			
	1	2	3	4
Entrega Continua				
EC1 Documentar estado actual de pruebas	■			
EC2 Definir alcance de pruebas unitarias por módulo	■			
EC3 Definir proceso para garantizar disponibilidad de datos de prueba		■		
EC4 Ejecutar pruebas en CI + publicar cobertura			■	
EC5 Garantizar umbral mínimo de cobertura (gate)			■	
EC6 Sesión de socialización (pruebas + datos)				■
Arquitectura				
ARQ1 Generar diagramas C4 de Contexto y Contenedores	■			
ARQ2 Inventario de stack con estado y riesgo	■			
ARQ3 Proceso ligero para elección de nuevas herramientas		■		
ARQ4 Sesión de socialización (documentos)				■
Gestión Lean y Monitorización - Monitorización Apps/Infra				
MON1 Documentar herramientas actuales de monitorización y alertas	■			
MON2 Estrategia de revisión periódica y visibilidad		■		
MON3 Pipelines de despliegue para métricas DORA		■		
MON4 Dashboard DORA				■
MON5 Monitoreo sintético de login			■	
MON6 Definir 1 SLI/SLO y alertas			■	
MON7 Sesión de socialización (pipelines y accesos)				■
Gestión Lean y Monitorización - Visualizar el trabajo				
VIS1 Estándar de documentación mínima	■			
VIS2 Proceso de calidad (diseño/requisitos/UI)				■
VIS3 Tablero Kanban estándar				■
VIS4 Visualización central (Confluence/Notion/Git)		■		
VIS5 Guías rápidas de mantenimiento de documentación			■	
VIS6 Ritual de revisión quincenal				■

Figura 3.9: Cronograma de Actividades Plan de Acción

3.7. Resumen del capítulo

En este capítulo se desarrolló un proceso de **diagnóstico y planificación** orientado a fortalecer la adopción de prácticas DevOps en *Hello Insight*. A partir de la aplicación del modelo de capacidades propuesto por Molina (2022), la socialización y priorización realizada junto con el equipo, y la recolección de **métricas DORA** (ene.–ago. 2025), fue posible establecer una línea base objetiva del estado actual y diseñar un **plan de acción de cuatro semanas** para abordar las brechas más críticas identificadas.

Síntesis del diagnóstico El diagnóstico evidenció que la organización se encuentra en un nivel inicial de madurez DevOps, con una clasificación global de **2.44/5** (Fig. 3.4), correspondiente a la categoría de *Oportunidades de mejora*. Los resultados mostraron que ninguna de las dimensiones evaluadas supera el umbral de 3.0 (Fig. 3.5). En particular, **Gestión Lean y Monitorización** presentó el promedio más bajo (2.20), mientras que **Entrega Continua** obtuvo el promedio más alto (2.63), aunque aún por debajo del nivel aceptable. Asimismo, el análisis de la distribución de capacidades reflejó un predominio de *oportunidades de mejora* en todas las dimensiones (Tab. 3.4), destacándose **Cultura** como la dimensión con mayor concentración de brechas (Fig. 3.6).

Priorización y foco Como resultado del proceso de socialización y revisión conjunta, se priorizaron cuatro capacidades clave (Tab. 3.5.1) con alto impacto técnico y estratégico: *Implementar*

gestión de datos de prueba, Utilizar arquitecturas de bajo acoplamiento, Monitorizar aplicaciones e infraestructura para decisiones de negocio y Visualizar el trabajo para comunicar calidad. La priorización combinó la línea base del diagnóstico, la alineación con los objetivos de la empresa y el consenso entre el Jefe de Producto y el equipo (Tab. 3.20).

Plan de acción (4 semanas) A partir de estas capacidades priorizadas, se diseñó un plan compuesto por **23 tareas**, con un esfuerzo total estimado de **74 horas** distribuidas a lo largo de cuatro semanas. El plan se organizó con una lógica incremental:

- **Semana 1:** levantamiento de información, definición de estándares y creación de artefactos base (estado de pruebas, diagramas C4, inventario tecnológico, documentación inicial de monitorización).
- **Semana 2:** definición de procesos y primeras automatizaciones (datos de prueba, criterios de selección de herramientas, pipelines DORA, visualización centralizada).
- **Semana 3:** implementación técnica y validación (CI con cobertura, *quality gate*, dashboard DORA, monitoreo sintético, definición de SLI/SLO, guías de mantenimiento).
- **Semana 4:** socializaciones, ajustes finales y cierre operativo (acuerdos del equipo, revisión de accesos, establecimiento de rituales, próximos pasos).

Riesgos y mitigaciones Durante la planificación se identificaron riesgos asociados a la ejecución del plan:

- **Subestimación del esfuerzo** en tareas técnicas clave: se propuso dividir actividades en entregables más pequeños (*slices*) y priorizar artefactos mínimos (documentos breves, prototipos funcionales).
- **Dependencias cruzadas** entre pipelines, alertas y accesos: se planteó asegurar la disponibilidad de *stakeholders* desde la semana 1 y establecer *checkpoints* semanales.
- **Baja adopción de prácticas** relacionadas con estándares y rituales: se recomendó reforzar socializaciones con acuerdos formales y asignar responsables claros (*owners*) por artefacto.

Cierre Este capítulo establece una **hoja de ruta viable y medible** para avanzar en la madurez DevOps de *Hello Insight*. Se logró pasar de un diagnóstico inicial fundamentado en datos (métricas DORA y evaluación de capacidades) a un plan de acción concreto, de corto alcance y con entregables verificables. El impacto de estas acciones será evaluado en el siguiente capítulo mediante la comparación entre la línea base y los resultados obtenidos tras la ejecución del plan, lo cual permitirá proyectar una segunda iteración enfocada en profundizar la automatización (IaC/GitOps), ampliar la cobertura de pruebas y fortalecer prácticas culturales habilitadoras.

Acción y Evaluación

Con el fin de dar continuidad al proceso iniciado tras la fase de diagnóstico y planificación, en este capítulo se presenta la etapa de *Acción y Evaluación*, correspondiente a las fases previas a la etapa de aprendizaje del ciclo iterativo de la metodología de investigación-acción. En esta fase se llevaron a cabo las intervenciones planificadas con base en los resultados del modelo de capacidades de Molina (2022) y se evaluaron nuevamente los niveles de madurez, con el propósito de reducir las brechas identificadas, avanzar hacia el estado deseado de madurez DevOps en el área de producto y establecer el estado final antes de la fase de aprendizaje (conclusiones).

Durante la fase de *Acción*, el objetivo consistió en ejecutar las actividades priorizadas en el plan consolidado (ver Tabla 3.22), validando su viabilidad dentro del contexto operativo de la organización, generando aprendizaje organizacional y promoviendo la adopción progresiva de prácticas DevOps orientadas a las capacidades seleccionadas. Las intervenciones se desarrollaron a lo largo de cuatro semanas, abordando las dimensiones de *Entrega Continua*, *Arquitectura* y *Gestión Lean y Monitorización*. Cada conjunto de actividades fue diseñado con un alcance incremental, permitiendo recolectar evidencias tangibles —documentación, workflows, tableros, actas y métricas— sin afectar la operación productiva.

Posteriormente, en la fase de *Evaluación*, se aplicó nuevamente el modelo de capacidades de Molina (2022) con el fin de medir el impacto de las acciones implementadas. Si bien las intervenciones se centraron en un conjunto específico de capacidades, sus efectos se extendieron hacia otras dimensiones del modelo, evidenciando impactos transversales del plan de acción en diferentes áreas de la práctica DevOps.

Esta segunda evaluación permitió comparar los resultados con la línea base obtenida en el diagnóstico inicial, identificando avances en las dimensiones intervenidas, aprendizajes derivados de la ejecución y nuevas oportunidades de mejora para ciclos posteriores. Asimismo, se observaron mejoras indirectas en aspectos relacionados con la colaboración, la cultura organizacional y la gestión del conocimiento técnico.

En este capítulo se describen, en primer lugar, las actividades ejecutadas durante la fase de acción para cada capacidad priorizada; posteriormente, el proceso de evaluación y los resultados obtenidos tras la segunda aplicación del modelo de capacidades de Molina; y, finalmente, una síntesis de los principales logros, dificultades y aprendizajes surgidos de esta iteración.

4.1. Ejecución Plan de Acción

4.1.1. Metodología de ejecución

La ejecución se desarrolló de manera iterativa, combinando sesiones técnicas con actividades de revisión y socialización. Para garantizar la trazabilidad entre el diagnóstico y las acciones implementadas, cada intervención mantuvo un *Definition of Done* (DoD) documentado en el repositorio de la documentación definido también en este plan de acción llamado *hi-engineering-handbook*.

El equipo involucrado estuvo compuesto por el área de Producto: gestión de proyectos, diseño y desarrollo. La coordinación de la ejecución se realizó mediante reuniones quincenales, donde se revisaron avances y bloqueadores.

4.1.2. Ejecución: Dimensión - Capacidad

4.1.2.1. Entrega Continua — Gestión de datos de prueba

Objetivo: Asegurar la disponibilidad y calidad de los datos de prueba, priorizando pruebas unitarias y escalando progresivamente a otros niveles.

Durante las dos primeras semanas se documentó el estado actual de las pruebas unitarias y de integración, identificando los frameworks en uso, el nivel de cobertura y los módulos con mayor déficit de pruebas. Posteriormente, se definió el alcance de las pruebas unitarias por repositorio (*hi-api*, *hi-frontend*, *hi-analytics*) y se establecieron criterios mínimos de cobertura. Esta documentación ahora hace parte del repositorio de documentación *hi-engineering-handbook* en la sección `/docs/tests` y en la definición del workflow *Build & Test* en la sección `/docs/ci-cd`.

En la tercera semana, se integró la ejecución automática de pruebas al flujo de integración continua mediante *workflows* de *GitHub Actions*, de modo que cada *pull request* ejecuta la batería de pruebas definida para el repositorio correspondiente y publica el resultado junto con un reporte de cobertura; como resultado de estas integraciones, se consolidó y documentó el flujo unificado de integración y entrega continua mostrado en la Figura 4.1, donde el esquema de colores indica el tipo de acción requerida en el pipeline: en **morado** se presentan los eventos de *GitHub* que disparan el proceso (p. ej., *pull request*, *push*, *release*), en **naranja** los *jobs* o tareas que deben completarse para permitir el avance del flujo, como el empaquetado y los tests, en **amarillo** los *steps* automáticos ejecutados por el pipeline y en **azul** los *steps* manuales que requieren intervención humana (como aprobaciones o verificaciones), finalizando con las acciones como comandos o instrucciones que determinan el éxito de cada *job*.

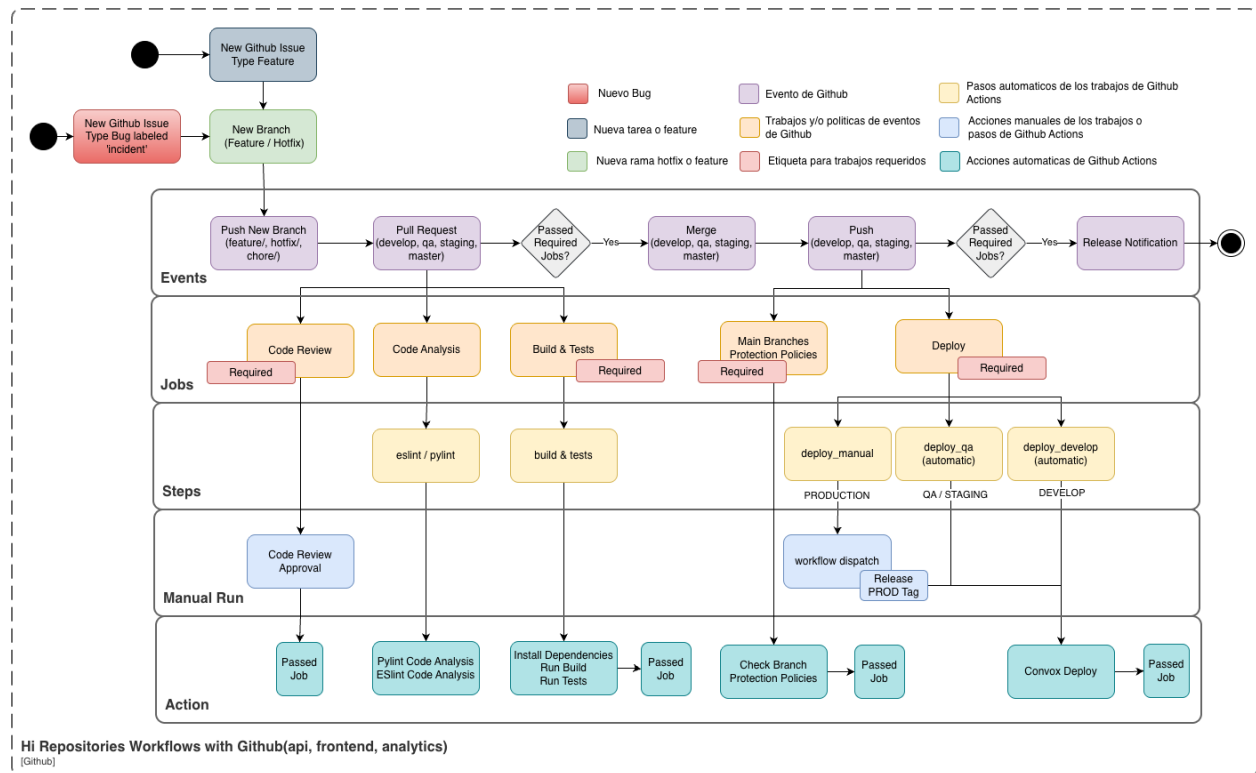


Figura 4.1: Diagrama General Procesos de CI/CD en Hello Insight

Durante esta etapa también se definió un proceso básico y repetible para garantizar la disponibilidad y viabilidad de los datos de prueba. Tal como señalan **Crispin and Gregory (2009)** en el Capítulo 15, la preparación anticipada de los datos y su verificación continua son actividades esenciales en equipos ágiles, pues permiten asegurar que cada ciclo de desarrollo cuente con la información necesaria para ejecutar pruebas de forma confiable. A partir de estas recomendaciones, el equipo consolidó un flujo liviano que incluye la identificación temprana de los datos requeridos, la validación de su consistencia y la mejora continua de los mecanismos para su generación y mantenimiento.

Como resultado, se obtuvo una visión consolidada del estado de las pruebas dentro del ecosistema de aplicaciones y se promovió la estandarización de esta práctica en el equipo. La socialización de estos avances permitió formalizar acuerdos sobre la creación, disponibilidad y mantenimiento de los datos de prueba en los entornos de desarrollo y QA.

4.1.2.2. Arquitectura — Bajo acoplamiento

Objetivo: Estandarizar el stack tecnológico para facilitar la escalabilidad y mantenibilidad de los sistemas.

Se elaboraron los diagramas C4 de contexto y contenedores, representando las principales interacciones entre servicios, bases de datos y componentes externos. Estos diagramas fueron almacenados en la carpeta `/docs/architecture/c4/` como base para futuras revisiones arquitectónicas.

Paralelamente se creó un inventario del stack tecnológico, identificando versiones, dependencias críticas y riesgos asociados a cada tecnología. Este ejercicio permitió visibilizar componentes obsoletos y dependencias que requerían actualización. En `/docs/stack/` quedó documentado el estado actual del stack tecnológico de *Hello Insight*.

Durante la segunda iteración se definió un proceso ligero para la elección de nuevas herramientas, apoyado en plantillas ADR (*Architecture Decision Records*), un mecanismo recomendado por Nygard (2011) para documentar de forma estructurada y concisa las decisiones arquitectónicas y el razonamiento que las sustenta. Este formato permitirá asegurar trazabilidad, claridad y consistencia en las decisiones técnicas del equipo. La encuesta utilizada para diligenciar y validar la información requerida por los ADR se presenta en el anexo C.

El cierre del ciclo incluyó una sesión de socialización de los documentos elaborados, en la que se acordó mantenerlos como referencia para futuras decisiones tecnológicas y como parte del repositorio de arquitectura en *Hello Insight*.

4.1.2.3. Gestión Lean y Monitorización — Aplicaciones e infraestructura

Objetivo: Medir la salud de las aplicaciones e infraestructura más allá de métricas técnicas, con enfoque en la experiencia de usuario y los indicadores de negocio.

La primera semana se destinó a documentar las herramientas de monitorización y alertas existentes, identificando coberturas y vacíos. En la segunda, se definieron estrategias de revisión periódica de dashboards y se consolidaron las responsabilidades de monitoreo.

Además de la documentación del estado de las herramientas de monitorización y de las estrategias de revisión periódica, durante esta etapa se avanzó en la formalización de prácticas orientadas a la fiabilidad del sistema mediante la definición de los primeros *Service Level Indicators* (SLI) y *Service Level Objectives* (SLO) para los servicios críticos de producción. Para ello, se consolidó un conjunto de indicadores basados en la disponibilidad y latencia de los servicios principales de *Hello Insight*, medidos mediante monitoreo sintético con UptimeRobot. Estos SLIs consideran puntos de verificación en los componentes clave del ecosistema—frontend, API, analytics, base de datos y encuestas—y permiten evaluar su disponibilidad sobre un periodo de 365 días. Con base en estos indicadores se estableció un SLO de disponibilidad mensual $\geq 99,99\%$, equivalente a un presupuesto de error aproximado de cinco minutos de inactividad por mes.

De forma complementaria, se implementaron monitores sintéticos para el proceso de autenticación, abarcando distintos roles funcionales del sistema, con el fin de evaluar la disponibilidad desde la perspectiva del usuario y no únicamente desde la infraestructura. Asimismo, se configuraron políticas de alerta y canales de notificación en tiempo real mediante Slack y correo electrónico, fortaleciendo la capacidad de respuesta del equipo ante incidentes operativos.

Para la medición de las métricas DORA, se integraron los repositorios de GitHub de *Hello Insight* con Apache DevLake, utilizando el flujo de extracción, transformación y visualización definido por la propia herramienta. En particular, se emplearon los dashboards predefinidos de DORA incluidos en DevLake, los cuales permiten calcular automáticamente la frecuencia de despliegue, el tiempo de entrega de cambios, la tasa de fallos y el tiempo medio de recuperación. Esta integración, basada en los mecanismos descritos por [Apache Software Foundation \(2023\)](#), proporcionó una vista consolidada y trazable del desempeño del proceso de entrega, habilitando un análisis objetivo y comparable en el tiempo.

En conjunto, estas acciones fortalecieron de manera significativa la observabilidad y la fiabilidad del ecosistema de aplicaciones de *Hello Insight*. La definición de *Service Level Indicators* (SLI) y *Service Level Objectives* (SLO), el uso de presupuestos de error, la instrumentación de monitoreo sintético para flujos críticos de usuario y la configuración de alertas orientadas al cumplimiento de SLOs son lineamientos descritos en [Beyer et al. \(2016\)](#). Estas prácticas, combinadas con la visibilidad obtenida a través de las métricas DORA y los tableros centralizados de observabilidad, permitieron al equipo adoptar un enfoque operativo más proactivo y basado en datos.

4.1.2.4. Gestión Lean y Monitorización — Visualización del trabajo

Objetivo: Generar documentación mínima y tableros visuales que permitan comunicar el estado de los proyectos y la calidad del software al equipo.

Durante esta capacidad se diseñó e implementó un conjunto de actividades orientadas a mejorar la transparencia operativa del trabajo del equipo de ingeniería. Como primer paso, se definió un estándar de documentación mínima, inspirado en prácticas de documentación ligera recomendadas por [Cockburn \(2001\)](#), el cual establece los elementos esenciales que deben contener todos los proyectos y todas las historias de usuario. En el caso de *Hello Insight*, estos elementos incluyen: (1) una descripción clara del propósito y contexto del proyecto; (2) el alcance inicial y las restricciones relevantes; (3) la identificación de actores y roles involucrados; (4) historias de usuario redactadas bajo un formato estandarizado con criterios de aceptación en Gherkin; (5) modelos livianos, como diagramas de arquitectura o flujos, cuando aportan claridad; (6) definiciones explícitas de *Definition of Ready* y *Definition of Done*; y (7) registros mínimos de decisiones técnicas mediante plantillas ADR. Todos estos elementos fueron documentados dentro del *hi-engineering-handbook* tras un proceso de análisis del flujo actual de gestión de proyectos en la organización, lo que permitió alinear prácticas, reducir inconsistencias y consolidar un marco operativo común para el equipo.

Posteriormente, se creó una Guía rápida de mantenimiento de documentación, incorporada igualmente en el *hi-engineering-handbook*, que formaliza principios y prácticas para garantizar que la documentación se mantenga actualizada. Esta guía adopta la noción propuesta por [Poppendieck and Poppendieck \(2003\)](#) de tratar la documentación como parte del flujo de valor y evitar la acumulación de “inventario” innecesario, promoviendo ciclos de actualización más cortos y continuos.

Para mejorar la visibilidad del flujo de trabajo, se definió un tablero Kanban estándar, fundamentado en los principios de visualización y control del flujo de [Anderson \(2010\)](#). Este tablero

incluye límites de trabajo en progreso (WIP), estados consensuados y reglas de avance basadas en los criterios de aceptación, el *Definition of Ready* y el *Definition of Done*. Su adopción permitió unificar criterios entre frontend, backend, analytics y DevOps, generando una perspectiva compartida del estado del trabajo y fortaleciendo la coordinación entre roles.

En paralelo, se consolidó *Notion* como herramienta central de visualización, escogida para servir como un punto de comunicación transversal para toda la organización. Esta visualización integra el estado de los proyectos, enlaces a documentación, decisiones arquitectónicas y referencias clave, complementando la información existente en los repositorios de GitHub —donde se gestionan, además del código, historias de usuario, bugs y tareas— para mantener una fuente unificada de verdad. Esta estrategia responde a la recomendación de Highsmith (2002) de mantener artefactos vivos, accesibles y continuamente actualizados, promoviendo una comprensión compartida del trabajo y evitando la dispersión de información entre equipos y áreas funcionales.

Adicionalmente, el trabajo desarrollado en el *hi-engineering-handbook* permitió consolidar un proceso de calidad que proporciona los elementos fundamentales para evaluar y asegurar la consistencia de los proyectos. Este proceso integra todo lo definido en la documentación mínima, criterios formales de *Definition of Ready* y *Definition of Done*, lineamientos para la revisión de diseño y requisitos, y listas de verificación técnicas asociadas a pruebas, documentación y cumplimiento funcional. A esto se sumaron mecanismos automáticos de verificación mediante los pipelines de integración continua, donde cada *pull request* ejecuta pruebas unitarias, validaciones de compilación y análisis de cobertura antes de ser elegible para integración. Asimismo, se estableció como regla la revisión obligatoria de al menos un miembro del equipo en cada solicitud de cambio, formalizando prácticas de *peer review* que favorecen la calidad, la coherencia técnica y la transferencia de conocimiento.

En su conjunto, estos elementos conforman un marco inicial de procesos de calidad del proyecto, integrando atributos como la completitud de requisitos, la claridad de los criterios de aceptación, la consistencia documental, la estabilidad de los entregables y la verificación continua mediante automatización y revisión por pares. Este marco también sienta las bases para la definición de indicadores clave de calidad, abarcando aspectos relacionados con la solución técnica, los requisitos funcionales y no funcionales, y la experiencia de usuario (UI). Aunque estos componentes representan una primera consolidación del proceso de calidad en *Hello Insight*, también habilitan una ruta estructurada para futuras iteraciones, en las cuales será posible extender este sistema hacia métricas más robustas, mecanismos formales de aseguramiento de calidad y criterios avanzados para monitorear el estado de los proyectos y reducir el retrabajo.

Finalmente, se estableció un ritual de revisión quincenal de documentación, siguiendo principios de mejora continua Lean descritos por Poppendieck and Poppendieck (2003). En este espacio, el equipo revisa activos clave, detecta obsolescencias, actualiza elementos críticos y registra acuerdos para mantener la documentación alineada con la evolución del producto y los procesos internos.

En conjunto, estas actividades fortalecieron significativamente la capacidad de *Hello Insight* para visualizar el trabajo, comunicar calidad y mantener información confiable en repositorios, tableros y canales de coordinación. Esto contribuyó directamente al aumento de madurez en la dimensión de Gestión Lean y Monitorización, consolidando prácticas de transparencia, estandarización y mejora

continua que sirven como base para iteraciones posteriores del proceso DevOps en la organización.

4.1.3. Cierre del plan de acción

La ejecución del plan de acción se completó dentro del horizonte de cuatro semanas, siguiendo una dinámica iterativa que combinó implementación técnica, documentación y espacios de socialización. Como resultado, se consolidaron artefactos y prácticas base en cuatro frentes: (i) estandarización y automatización de pruebas en integración continua, (ii) fortalecimiento de la documentación arquitectónica y del inventario del stack, (iii) formalización de prácticas de observabilidad y fiabilidad mediante SLI/SLO, monitoreo sintético y alertas, y (iv) estandarización de mecanismos de visualización del trabajo y control de calidad a través de tableros y documentación mínima.

Los resultados que se presentan a continuación deben interpretarse como **resultados preliminares** centrados en evidencias de implementación y adopción inicial. La medición del impacto en desempeño (mejoras verificables en madurez) se realizará en la siguiente fase del estudio mediante la segunda evaluación, donde se aplicará nuevamente el modelo de capacidades.

4.1.4. Resultados preliminares del plan de acción

Los resultados preliminares se estructuran como evidencia directa de ejecución, enfocada en los cambios incorporados en repositorios, flujos de trabajo y documentación. En conjunto, el plan permitió aumentar la trazabilidad entre diagnóstico, acciones y entregables, y dejó instalados mecanismos para sostener la mejora continua.

Entre los logros preliminares más relevantes se destacan:

- **Integración de pruebas y cobertura en CI:** se incorporaron ejecuciones automáticas de pruebas y reportes de cobertura visibles en *pull requests*, junto con criterios mínimos documentados por repositorio en *hi-engineering-handbook*.
- **Base documental de arquitectura y stack:** se generaron y centralizaron diagramas C4 (contexto y contenedores), así como un inventario del stack tecnológico con dependencias críticas y riesgos, habilitando revisiones arquitectónicas futuras.
- **Observabilidad y fiabilidad:** se definieron SLIs/SLOs iniciales, se instrumentó monitoreo sintético para flujos críticos (incluyendo autenticación por roles) y se configuraron alertas y canales de notificación para mejorar la respuesta ante incidentes.
- **Estandarización de procesos y visualización del trabajo:** se formalizaron estándares de documentación mínima, DoR/DoD, reglas de avance en Kanban y guías de mantenimiento documental.

De manera complementaria, se identificaron retos preliminares asociados a la adopción sostenida de las prácticas, como la necesidad de reforzar la disciplina de actualización continua de la documentación, así como garantizar la calidad y consistencia de los datos para las métricas (DORA y SLO). Estos aspectos serán considerados en la fase de evaluación, mediante una segunda

aplicación del modelo de capacidades de Molina, para determinar en qué medida las prácticas se internalizaron y si se reflejan en cambios medibles de madurez DevOps.

4.2. Evaluación - Reassessment (Segunda aplicación del modelo)

Esta sección describe la implementación de la segunda aplicación del modelo de capacidades de Molina: *Reassessment*. Su objetivo es recolectar nuevamente la información a través de la encuesta, con el fin de comparar los resultados frente al diagnóstico inicial y determinar, de forma verificable, si las acciones ejecutadas generaron algún tipo de impacto en la madurez DevOps de la empresa *Hello Insight*. Se mantuvieron las mismas definiciones de capacidades, la misma escala de medición y un procedimiento de aplicación consistente con el utilizado en la primera evaluación.

4.2.1. Preparación

4.2.1.1. Comunicación

Previo a la apertura de la encuesta, se realizó una comunicación formal al equipo con el propósito de explicar el objetivo del *reassessment*, el alcance de la medición y la importancia de responder desde la experiencia real posterior a la ejecución del plan de acción. La comunicación incluyó el periodo de evaluación y los lineamientos para asegurar consistencia en las respuestas (por ejemplo, responder con base en evidencias observables como documentación, flujos de CI/CD, tableros y prácticas operativas). Adicionalmente, se informó la fecha de apertura y cierre, y se habilitó un canal para resolver dudas durante la ventana de recolección.

4.2.1.2. Encuestados

Los siguientes miembros del equipo de *Hello Insight* fueron encuestados nuevamente en atención a su rol dentro del área de producto, con el propósito de obtener una medición comparable con la evaluación inicial y capturar la percepción posterior a la ejecución del plan de acción. De esta manera, las respuestas de estos participantes constituyen la base para contrastar cuantitativamente los resultados del *diagnóstico* (antes) y del *reassessment* (después), identificando variaciones en los niveles de capacidad y en los promedios por dimensión.

- Jefe de Producto (CPO)
- Diseñador
- Líder Técnico
- Desarrollador Backend Sr.
- Desarrollador Frontend Sr.
- Desarrollador Frontend Jr.

De la lista inicial, fueron excluidos los siguientes miembros del equipo.

- Gerente de Proyecto (PM) - **Justificación:** El miembro del equipo fue retirado de la empresa.
- Director de Investigación - **Justificación:** El miembro del equipo no estuvo involucrado en las actividades realizadas en el plan de acción.

4.2.1.3. Plataforma

Para el *reassessment* se reutilizó la misma encuesta ya montada para el diagnóstico inicial en Microsoft Forms, preservando la estructura del cuestionario, las preguntas, las opciones de respuesta y la configuración general, con el fin de garantizar la comparabilidad entre ambas mediciones. Gracias a la integración de la plataforma, se facilitó la recolección y consolidación de los resultados mediante su vinculación a un archivo de Microsoft Excel.

4.2.2. Ejecución

4.2.2.1. Encuesta

La ejecución consistió en la aplicación del instrumento del modelo de capacidades nuevamente a los participantes definidos. Cada encuestado respondió de manera individual, evaluando el nivel alcanzado por cada capacidad con base en su experiencia posterior a la implementación del plan de acción.

4.2.2.2. Apertura y cierre de la encuesta

La encuesta fue habilitada durante una ventana de dos(2) semanas. En la apertura se compartió el enlace oficial junto con las instrucciones de diligenciamiento. Se estableció un recordatorio intermedio para incrementar la tasa de respuesta y se comunicó la fecha de cierre con antelación. Una vez finalizado el periodo, se cerró el formulario y se exportaron los resultados para su análisis. El conjunto de respuestas obtenido se utilizó como insumo para presentar los resultados que encontramos en la siguiente sección.

4.2.3. Resultados

4.2.3.1. Transformación en unidades de análisis

Para el *reassessment* se aplicaron los mismos mecanismos de transformación y análisis utilizados en el diagnóstico inicial, con el fin de garantizar la comparabilidad entre ambas mediciones. En particular, se dispuso nuevamente de dos archivos obtenidos en las etapas previas: (i) el consolidado de preguntas, respuestas, planes de acción y demás información, denominado *Base de Datos Encuesta* (`bd_encuestas.csv`), descargado en formato CSV; y (ii) el archivo con las respuestas diligenciadas por los participantes en esta segunda medición, denominado *Respuestas* (`respuestas.csv`).

También se reutilizó el script de limpieza y *scoring* empleado en la evaluación inicial, ajustando únicamente la fuente de entrada para apuntar al archivo de respuestas del *reassessment*. Al igual

que en el diagnóstico, previo a la etapa final del procesamiento se eliminaron las opciones de respuesta *No aplica* y *No sabe/No responde* de cada pregunta, con el fin de omitir del análisis aquellas alternativas que no están alineadas con el modelo y centrarse únicamente en los elementos definidos en la escala propuesta.

Finalmente, se mantuvo la misma estrategia de análisis basada en seleccionar el valor de menor denominación entre las respuestas de cada ítem. Por ejemplo, si en una pregunta las respuestas varían entre 4, 5 y 1, el valor seleccionado es 1. Esta decisión metodológica se conservó para identificar brechas potenciales de raíz y asegurar consistencia en la comparación antes/después. El resultado del procesamiento se consolidó en las mismas estructuras de salida definidas en el diagnóstico, permitiendo el cálculo de promedios por capacidad y dimensión, así como la clasificación de resultados.

Para la interpretación de los resultados se utilizaron los mismos puntos de referencia (*benchmark*) definidos por Molina (2022), los cuales clasifican las capacidades evaluadas según el valor alcanzado en las respuestas.

En la siguiente sección se presenta el reporte de los resultados del *reassessment* con su respectivo *delta*, para evidenciar el cambio que tuvo en el nivel de madurez el plan de acción implementado en **Hello Insight**

4.2.3.2. Reporte

Esta sección presenta los resultados de la evaluación posterior a la ejecución del plan de acción (*reassessment*) y su comparación con el diagnóstico inicial. El análisis se organiza en tres niveles: (i) resultados por ítem (capacidad), (ii) comparación diagnóstico vs. *reassessment* mediante *delta*, y (iii) síntesis por clasificación, por dimensión y resultado global, apoyada en las Figuras 4.3, ?? y 4.6.

Resultados por ítem (capacidad). La Tabla 4.1 consolida el resultado del *reassessment* para cada pregunta del instrumento, indicando la capacidad evaluada, el puntaje obtenido y su clasificación frente a los puntos de referencia (Fortalezas, Aceptable u Oportunidades de mejora). En términos generales, el *reassessment* evidencia un estado favorable en la mayoría de capacidades, con especial concentración en categorías superiores.

ID Pregunta	Capacidad	Resultado	Calificación
Q1	Utilizar control de versiones para todos los artefactos de producción	4	Fortalezas
Q2	Aumentar el proceso de despliegue	4	Fortalezas
Q3	Implementar integración continua	3	Aceptable
Q4	Utilizar métodos de desarrollo “Trunk-Based”	4	Fortalezas
Q5	Implementar pruebas automáticas	3	Aceptable

ID Pregunta	Capacidad	Resultado	Calificación
Q6	Implementar gestión de los datos de prueba	3	Aceptable
Q7	Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)	3	Aceptable
Q8	Implementar entrega continua de software	4	Fortalezas
Q9	Utilizar arquitecturas de bajo acoplamiento	4	Fortalezas
Q10	Permitir a los equipos definir sus propias arquitecturas	4	Fortalezas
Q11	Recolectar e implementar las recomendaciones (feedback) de los clientes	3	Aceptable
Q12	Hacer visible el flujo de trabajo durante todo el ciclo	2	Oportunidades de mejora
Q13	Trabajar con lotes o cantidades pequeñas	4	Fortalezas
Q14	Fomentar y habilitar al equipo para realizar experimentos	3	Aceptable
Q15	Tener un proceso liviano para la aprobación de cambios	1	Oportunidades de mejora
Q16	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	3	Aceptable
Q17	Monitorizar la salud de los servicios y aplicaciones proactivamente	3	Aceptable
Q18	Administrar y establecer límites para el trabajo en progreso (Work In-Progress)	4	Fortalezas
Q19	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	4	Fortalezas
Q20	Implementar una cultura tipo generativa	4	Fortalezas
Q21	Animar y apoyar al equipo a aprender	4	Fortalezas
Q22	Apoyar y facilitar la colaboración entre equipos	5	Fortalezas

ID Pregunta	Capacidad	Resultado	Calificación
Q23	Apoyar y facilitar la colaboración entre equipos	4	Fortalezas
Q24	Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos	5	Fortalezas
Q25	Apoyar el liderazgo transformacional	4	Fortalezas

Tabla 4.1: Resultados obtenidos tras el *reassessment*

Comparación diagnóstico vs. *reassessment* (*delta*). Con el fin de mantener consistencia con el diagnóstico inicial, se conservó la misma estrategia metodológica: para cada ítem se seleccionó el valor de menor denominación entre las respuestas del grupo encuestado, de modo que el análisis refleje la brecha más crítica observada en el equipo. Bajo este enfoque, la Tabla 4.2 presenta, por pregunta, el resultado del diagnóstico, el resultado del *reassessment* y el *delta* calculado como $\Delta = \text{Evaluación} - \text{Diagnóstico}$.

ID Pregunta	Diagnóstico	Evaluación	Δ (Eval - Diag)
Q1	4	4	0
Q2	1	4	3
Q3	3	3	0
Q4	3	4	1
Q5	3	3	0
Q6	2	3	1
Q7	3	3	0
Q8	2	4	2
Q9	1	4	3
Q10	4	4	0
Q11	2	3	1
Q12	2	2	0
Q13	3	4	1
Q14	3	3	0
Q15	1	1	0
Q16	2	3	1
Q17	3	3	0
Q18	4	4	0
Q19	1	4	3
Q20	2	4	2
Q21	3	4	1

ID Pregunta	Diagnóstico	Evaluación	Δ (Eval - Diag)
Q22	2	5	3
Q23	3	4	1
Q24	2	5	3
Q25	2	4	2

Tabla 4.2: Comparación de resultados por ítem entre diagnóstico y *reassessment*, con delta por pregunta

Como síntesis, se observa que 15 de las 25 capacidades evaluadas presentaron mejoras ($\Delta > 0$) y 10 se mantuvieron estables ($\Delta = 0$). No se registraron retrocesos: ninguna capacidad presentó un delta negativo ($\Delta < 0$), teniendo en cuenta que una de las capacidades estables se ubica en la categoría más baja con el peor resultado: Oportunidad de Mejora y 1. En conjunto, este patrón sugiere que las acciones implementadas fortalecieron prácticas existentes sin generar degradación en las capacidades previamente consolidadas.

Impacto sobre capacidades priorizadas (prioridad 1). Para aislar el efecto del plan de acción sobre las capacidades intervenidas de manera directa, la Tabla 4.3 resume el comportamiento de las capacidades definidas como **prioridad 1**, comparando diagnóstico vs. *reassessment* y su respectivo *delta*.

ID	Capacidad (prioridad 1)	Diagnóstico	Evaluación	Δ
Q6	Implementar gestión de los datos de prueba	2	3	1
Q9	Utilizar arquitecturas de bajo acoplamiento	1	4	3
Q16	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	2	3	1
Q19	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	1	4	3

Tabla 4.3: Comparación diagnóstico vs. *reassessment* para capacidades priorizadas (prioridad 1)

Los resultados evidencian mejoras en las cuatro capacidades intervenidas directamente. Dos capacidades presentan incrementos moderados ($\Delta = 1$) asociados a avances incrementales y estabilización de prácticas; mientras que dos capacidades muestran incrementos significativos ($\Delta = 3$), lo cual indica un cambio material respecto al punto de partida. Este comportamiento respalda la pertinencia de la priorización definida en el plan de acción y su contribución al fortalecimiento de la madurez DevOps.

Distribución por clasificación. Al considerar la clasificación del *reassessment* (Tabla 4.1), se obtiene la siguiente distribución: **15** capacidades en *Fortalezas*, **8** en *Aceptable* y **2** en *Oportunidades de mejora*. La Figura 4.2 complementa esta lectura al mostrar el **movimiento neto** de capacidades

entre categorías desde el diagnóstico hacia la evaluación; posteriormente, la Figura 4.3 presenta el resultado desde una perspectiva agregada, evidenciando el cambio en el conteo final por categoría.

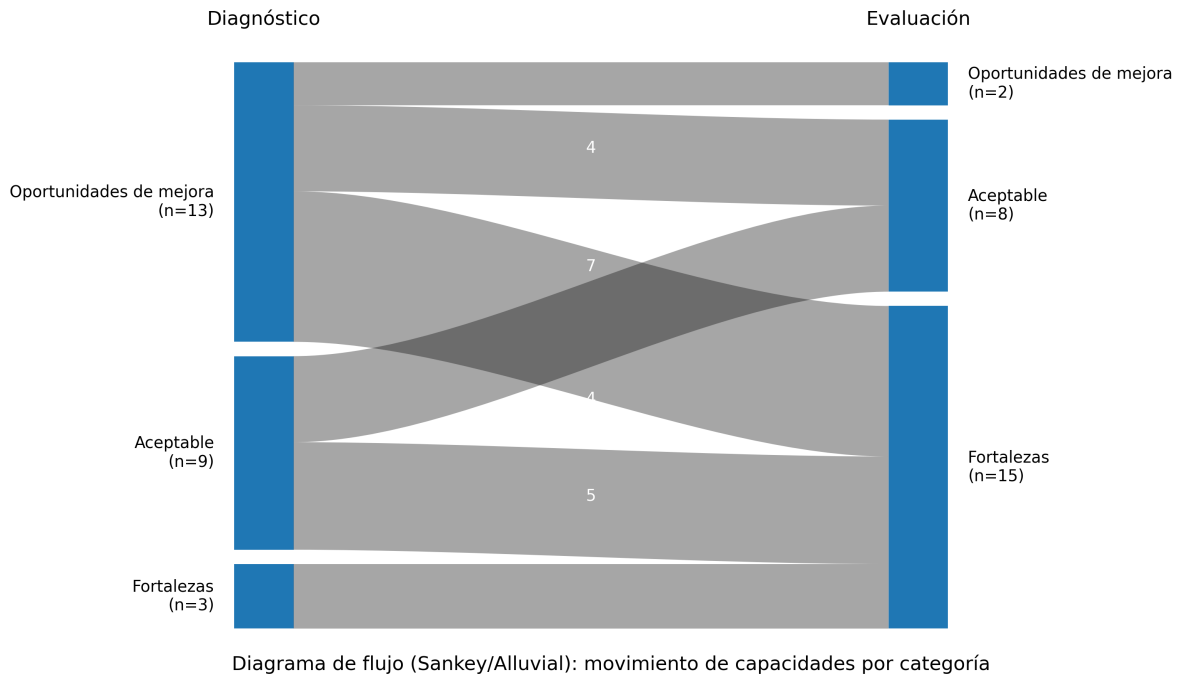


Figura 4.2: Diagrama de flujo (Alluvial): transición de capacidades entre categorías (diagnóstico → evaluación)

El diagrama Alluvial (Figura 4.2) muestra de forma explícita **cómo se redistribuyeron las capacidades** entre categorías desde el diagnóstico hacia la evaluación. En particular, se observa que la mayor parte del movimiento proviene de *Oportunidades de mejora* y *Aceptable* hacia *Fortalezas*, lo cual es consistente con el patrón reportado en los deltas (ausencia de retrocesos). Esta lectura es útil para la discusión porque evidencia que la mejora no se explica únicamente por incrementos marginales, sino por un **cambio estructural** en la distribución: se reducen capacidades que actúan como restricciones (*limitantes*) y se incrementan capacidades que estabilizan el sistema (*habilitadoras*). Desde la perspectiva del negocio, este desplazamiento sugiere menor fricción para liberar cambios, mayor estabilidad percibida del servicio y una reducción del trabajo reactivo (incidentes, retrabajo y coordinación *ad-hoc*), lo que preserva capacidad del equipo para ejecutar el *roadmap*.

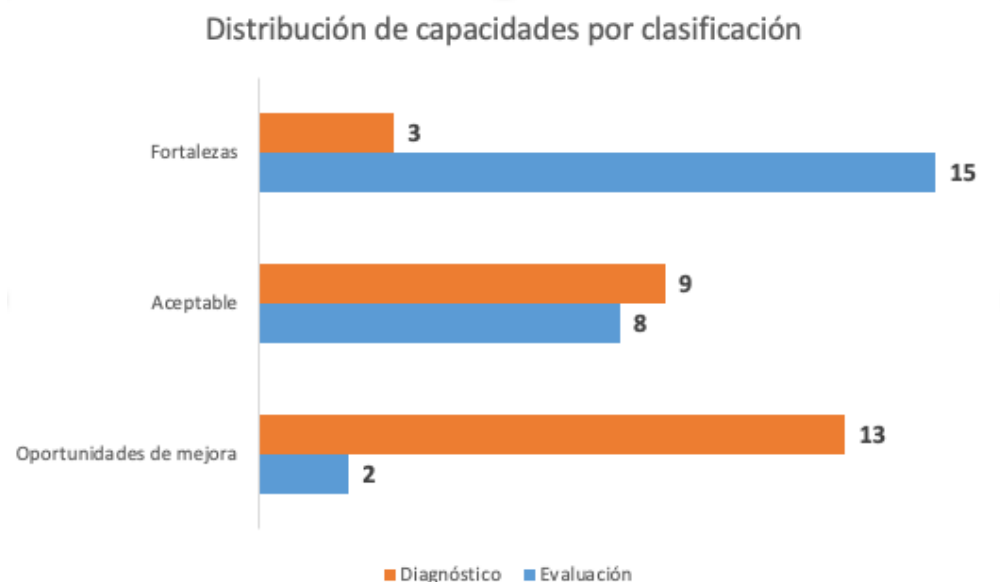


Figura 4.3: Distribución de capacidades según clasificación durante el diagnóstico y la evaluación

La Figura 4.3 confirma el cambio agregado en la distribución por categorías: en el diagnóstico se registraron **13** capacidades en *Oportunidades de mejora*, **9** en *Aceptable* y **3** en *Fortalezas*; mientras que en la evaluación posterior la distribución se desplazó a **2** en *Oportunidades de mejora*, **8** en *Aceptable* y **15** en *Fortalezas*. En términos prácticos, reducir la concentración de capacidades en el nivel más bajo disminuye la probabilidad de cuellos de botella y fallas tardías, y favorece una operación más predecible; por ello, este resultado complementa la mejora observada por dimensión y la evolución del puntaje global reportado en la Figura 4.6.

Resultados por dimensión: consistencia del avance. Además del análisis por capacidad, se evaluó el comportamiento por dimensión para verificar si las mejoras se concentraron únicamente en los frentes intervenidos o si se observó un avance transversal. La Figura 4.4 presenta el comparativo de puntajes por dimensión entre diagnóstico y *reassessment* en un formato de barras, mientras que la Figura 4.5 muestra el mismo resultado en un radar para facilitar la lectura del patrón global de mejora. En conjunto, se evidencia un **crecimiento en el promedio de todas las dimensiones**: las dimensiones intervenidas mostraron incrementos, como era esperable por el foco del plan; sin embargo, también se observaron **efectos secundarios** en dimensiones no abordadas de forma explícita, lo que sugiere que la estandarización de prácticas, la automatización y la mayor visibilidad operativa generaron un impacto positivo en capacidades habilitadoras. En términos de clasificación, este patrón se tradujo en un **desplazamiento general** desde *Oportunidades de mejora* hacia niveles superiores, ubicando las dimensiones en rangos *Aceptables* e incluso con componentes que alcanzan *Fortalezas*, lo cual respalda que el plan produjo un avance consistente y no aislado dentro del modelo.

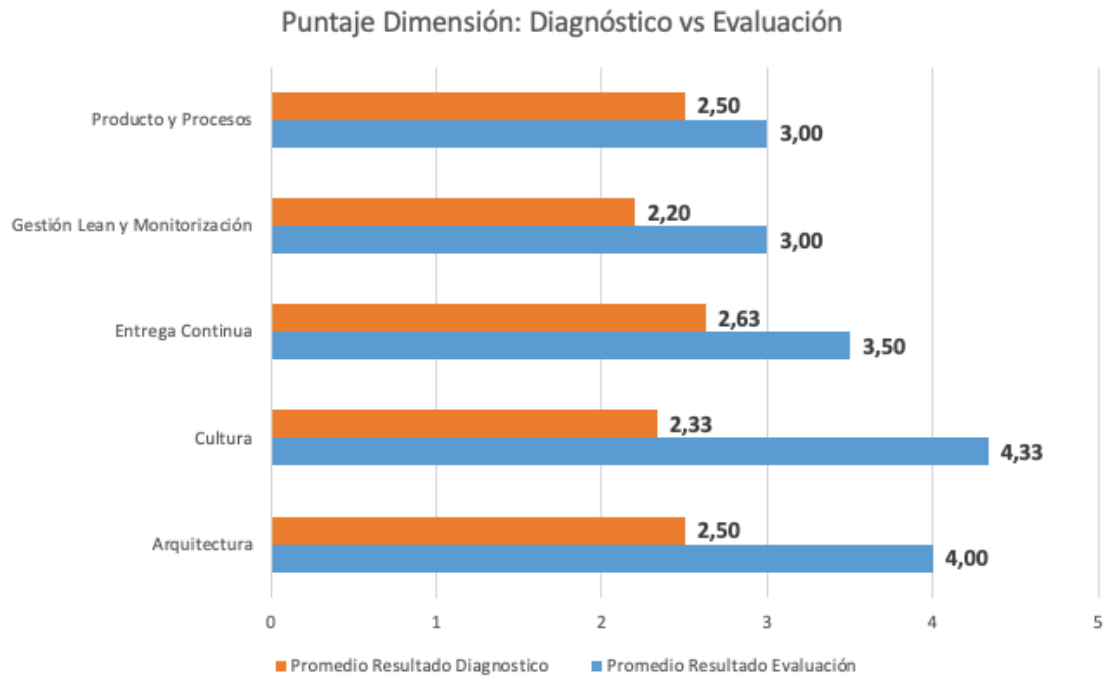


Figura 4.4: Comparativo por dimensión (barras): diagnóstico vs. evaluación

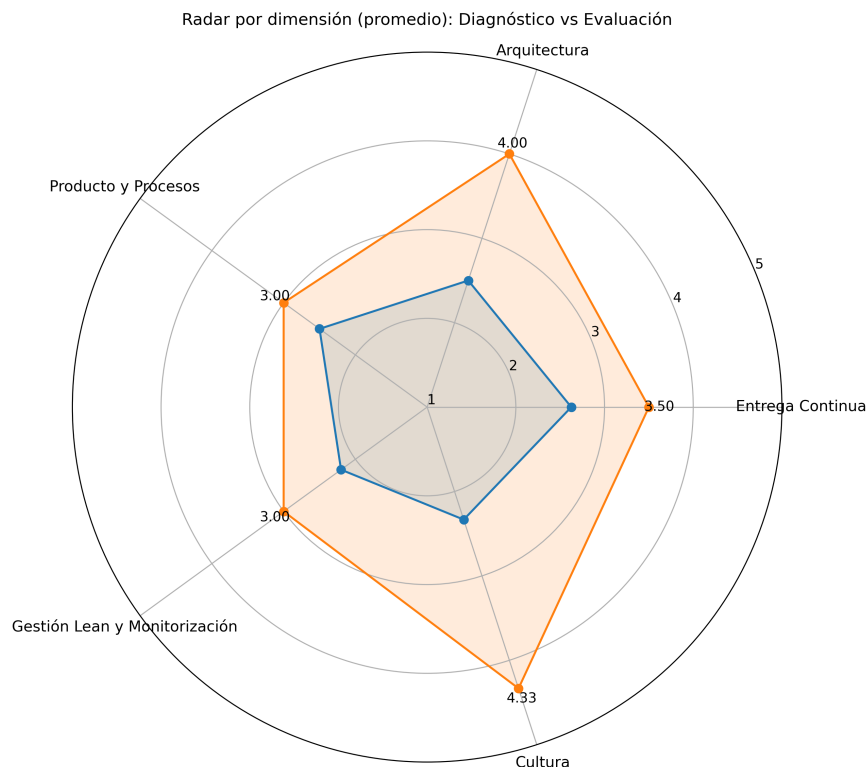


Figura 4.5: Radar por dimensión (promedio): diagnóstico vs. evaluación

En términos cuantitativos, *Entrega Continua* pasó de **2.63 a 3.50** ($\Delta = +0,88$), *Arquitectura* de **2.50 a 4.00** ($\Delta = +1,50$), *Producto y Procesos* de **2.50 a 3.00** ($\Delta = +0,50$) y *Gestión Lean y Monitorización* de **2.20 a 3.00** ($\Delta = +0,80$). Finalmente, *Cultura* presentó el mayor incremento, de **2.33 a 4.33** ($\Delta = +2,00$), lo que refuerza la interpretación de un efecto emergente sobre prácticas colaborativas y de responsabilidad compartida derivado de la automatización y la transparencia introducidas en el ciclo de entrega (Kim et al., 2021).

Explicación del crecimiento en la dimensión Cultura. Aunque la dimensión *Cultura* no fue definida como frente principal de intervención en el plan de acción, el incremento observado puede interpretarse como un efecto emergente derivado de la implementación de capacidades técnicas y orientadas a proceso. En particular, prácticas como la estandarización del pipeline de CI/CD, la automatización de validaciones (por ejemplo, ejecución de pruebas y reportes de cobertura por *pull request*) y la formalización de acuerdos operativos mediante documentación accesible, tienden a modificar el trabajo cotidiano al incrementar la transparencia, acortar los ciclos de retroalimentación y promover responsabilidad compartida por la calidad (Kim et al., 2021). Esta lectura se alinea con la evidencia reportada por DORA, donde se describe la *cultura generativa* como un habilitador clave del desempeño, debido a que favorece la cooperación, el flujo de información y el aprendizaje continuo (DevOps Research and Assessment (DORA), 2023).

Desde la perspectiva de gestión del cambio, Kotter plantea que los cambios se consolidan cuando los nuevos comportamientos se institucionalizan y se anclan en la forma normal de operar, especialmente cuando los sistemas y procesos refuerzan de manera consistente dichas conductas. Bajo este marco, los *gates* de calidad, los *checks* automatizados y la visibilidad sistemática de resultados en el pipeline funcionan como mecanismos de refuerzo que reducen la dependencia de acuerdos informales, estabilizan prácticas colaborativas y facilitan la adopción sostenida de nuevas rutinas (Kotter, 1996). En consecuencia, aun cuando el plan de acción se concentró en capacidades de carácter técnico-operativo, sus resultados pueden extenderse a la dimensión cultural al transformar el contexto en el que el equipo coordina, decide y aprende.

Efecto observado en Producto y Procesos. De manera similar, la mejora leve en la dimensión *Producto y Procesos* puede explicarse por la cercanía de varias actividades del plan con capacidades típicas de esta dimensión, tales como documentación mínima operativa, clarificación de criterios y reducción de ambigüedad en el flujo. Estos elementos suelen impactar positivamente la gestión del trabajo y la coordinación end-to-end, aun cuando no hayan sido el foco explícito de intervención (Forsgren et al., 2018).

Resultado global: cambio en la madurez total. Finalmente, para sintetizar el efecto del plan de acción a nivel agregado, la Figura 4.6 presenta el resultado total del diagnóstico vs. la evaluación. Se evidencia un incremento de **2.44 a 3.57**, lo cual representa una mejora neta de **+1.13** puntos en la escala de 1 a 5. Este aumento es consistente con la ausencia de deltas negativos, el desplazamiento hacia categorías superiores y las mejoras observadas en las capacidades priorizadas, consolidando evidencia cuantitativa del impacto del plan de acción.

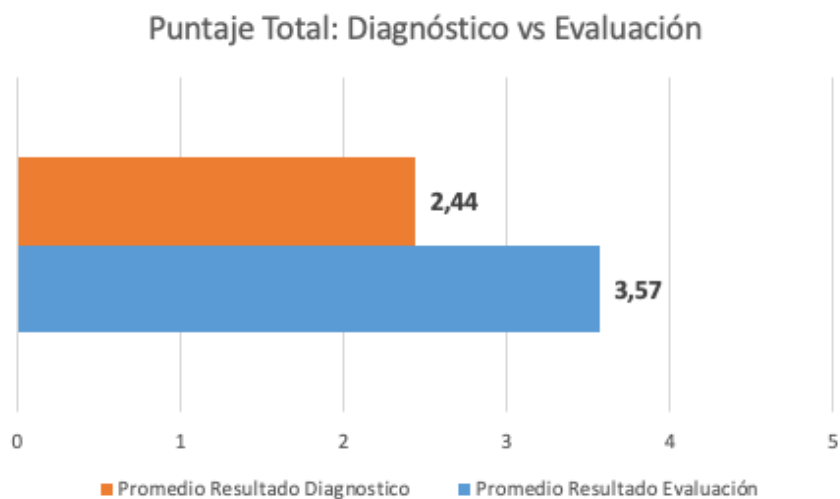


Figura 4.6: Gráfico comparativo del resultado total del diagnóstico vs. la evaluación

Implicaciones para el siguiente ciclo de mejora. Con el fin de mantener la misma metodología aplicada en el primer ciclo, se seleccionó nuevamente un conjunto de cuatro capacidades como prioridad 1 para el siguiente ciclo. La selección consideró criterios equivalentes a los utilizados previamente: (i) evidencia del *reassessment* sobre brechas críticas vigentes, (ii) comportamiento del *delta* para identificar estancamientos relevantes, y (iii) alineación con el objetivo de continuar consolidando la madurez DevOps, profundizando en monitoreo e incorporando progresivamente aspectos de seguridad dentro del ciclo de entrega.

En primer lugar, el *reassessment* identificó únicamente dos capacidades clasificadas como *Oportunidades de mejora*: **Q12** (hacer visible el flujo de trabajo durante todo el ciclo, resultado 2) y **Q15** (tener un proceso liviano para la aprobación de cambios, resultado 1). Estas capacidades se priorizaron debido a que representan restricciones sistémicas para el trabajo *end-to-end*: la visibilidad limitada del flujo dificulta gestionar dependencias, bloqueos y tiempos de espera, afectando la predictibilidad; mientras que la ausencia de un mecanismo liviano y explícito para la aprobación de cambios incrementa fricción y variabilidad en la promoción de cambios, debilitando la gobernanza de la entrega. En consecuencia, su permanencia en el nivel más bajo tras la ejecución del plan sugiere que constituyen el principal factor limitante para continuar elevando la madurez de manera sostenible.

En segundo lugar, para complementar el alcance del ciclo sin dispersar el foco, se seleccionaron dos capacidades adicionales por su contribución al propósito de profundizar la dimensión de monitoreo e iniciar la incorporación sistemática de seguridad dentro del enfoque DevOps. Bajo estos criterios, se incluyeron **Q17** (monitorizar la salud de los servicios y aplicaciones proactivamente) y **Q7** (incorporar los aspectos de seguridad en las fases de diseño y pruebas, *shift-left*). Aunque ambas capacidades se ubicaron en nivel *Acceptable* (resultado 3), el hecho de no presentar avance respecto al diagnóstico ($\Delta = 0$) indica un estancamiento y, por tanto, una oportunidad clara de maduración, coherente con los objetivos perseguidos desde la priorización inicial. Q17 se considera prioritaria porque el monitoreo proactivo y la observabilidad constituyen pilares de operación confiable, detección temprana y reducción de tiempos de recuperación, lo cual ayuda a sostener la estabilidad del sistema a medida que aumenta la velocidad de entrega (Beyer et al., 2016; Kim et al., 2021). Por su parte, Q7 se incorpora para extender el enfoque hacia DevSecOps, integrando controles y validaciones de seguridad desde etapas tempranas del ciclo de vida con el fin de reducir retrabajo, vulnerabilidades y fallas tardías; este enfoque se alinea con estándares y modelos de madurez que recomiendan desplazar la seguridad a la izquierda e integrarla dentro del pipeline (Committee, 2021; OWASP Foundation, 2025).

En síntesis (Tabla 4.4), el siguiente ciclo quedó estructurado con dos frentes complementarios: (i) remover los cuellos de botella principales identificados por el *reassessment* (Q12, Q15) y (ii) consolidar capacidades habilitadoras sin mejora observada para sostener y ampliar la madurez DevOps en monitoreo y seguridad (Q17, Q7).

ID	Capacidad (prioridad 1 para siguiente ciclo)	Diagnóstico	Evaluación	Δ
Q12	Hacer visible el flujo de trabajo durante todo el ciclo	2	2	0
Q15	Tener un proceso liviano para la aprobación de cambios	1	1	0
Q17	Monitorizar la salud de los servicios y aplicaciones proactivamente	3	3	0
Q7	Incorporar los aspectos de seguridad en las fases de diseño y pruebas (<i>shift-left</i>)	3	3	0

Tabla 4.4: Capacidades seleccionadas como prioridad 1 para el siguiente ciclo de mejora

4.2.4. Impacto en el negocio de Hello Insight

Como continuación del comparativo diagnóstico vs. *reassessment*, este apartado analiza el **impacto en el negocio** que tuvo la ejecución del plan de acción en *Hello Insight*. Para interpretar dicho impacto se utiliza como marco el *modelo de capacidades* aplicado en este estudio, el cual define qué se entiende por *capacidad* y cómo la mejora de capacidades relacionadas con entrega y operación se asocia con reducción de fricción y riesgo en la entrega de valor (Molina, 2022). Dado el contexto operativo de la empresa —equipo reducido en el área de Producto y ejecución con responsabilidades múltiples—, el incremento de madurez observado no se interpreta únicamente como un avance metodológico, sino como una reducción tangible de fricción operativa y de riesgo en la entrega de valor; en consecuencia, se espera una operación más estable y una ejecución más consistente del *roadmap* con el mismo tamaño de equipo.

En términos operativos, fortalecer capacidades asociadas a integración y entrega continua, estandarización de arquitectura, monitoreo/observabilidad y visualización del trabajo reduce la variabilidad del proceso y el “trabajo invisible” (coordinación *ad-hoc*, revisiones manuales repetidas, reprocesos por integración tardía o falta de trazabilidad). Al existir *workflows* más uniformes, criterios de calidad verificables antes de integrar cambios y un repositorio documental accesible, disminuye la dependencia de pasos manuales y de conocimiento tribal. Para el negocio, esto se traduce en **mayor predictibilidad** para planear *releases*, menor probabilidad de comprometer fechas por bloqueos inesperados y una ejecución más estable en periodos de alta demanda, lo cual impacta directamente la capacidad de cumplir compromisos con clientes y partes interesadas internas.

Adicionalmente, las mejoras en prácticas de calidad y operación incrementan la **confiabilidad percibida** del servicio y reducen el **costo de fallas**. La disponibilidad de señales operativas consistentes (*SLIs/SLOs*, monitoreo sintético y alertas) permite detectar degradaciones con mayor oportunidad y responder con criterios claros, evitando que incidentes o regresiones escalen a interrupciones prolongadas para usuarios. Desde el negocio, esto significa menos afectaciones en flujos críticos, menor carga de soporte reactivo y menos interrupciones del equipo de ingeniería para correcciones urgentes, preservando capacidad para entregar iniciativas de producto.

Finalmente, este impacto es especialmente relevante en *Hello Insight* porque la multi-asignación de roles amplifica cualquier ineficiencia: cuando las mismas personas alternan entre desarrollo,

validación, soporte y operación, cada incidente, retrabajo o dependencia de un experto concentra el riesgo y reduce la capacidad efectiva del equipo. En ese sentido, la estandarización (documentación mínima, trazabilidad técnica, reglas explícitas de flujo) y los mecanismos de coordinación visibles (tableros y acuerdos operativos) disminuyen cuellos de botella y aumentan resiliencia ante ausencias o rotación, lo cual permite sostener el desempeño de entrega en el tiempo.

4.3. Resumen del capítulo

En este capítulo se desarrolló la etapa de **Acción y Evaluación** como continuación del proceso iniciado tras el diagnóstico y la planificación. En primer lugar, se ejecutó el **plan de acción de cuatro semanas** definido a partir del modelo de capacidades de [Molina \(2022\)](#), manteniendo trazabilidad entre diagnóstico, actividades y entregables mediante evidencias verificables (documentación, *workflows*, tableros y artefactos técnicos) centralizadas en el repositorio *hi-engineering-handbook*. En segundo lugar, se realizó una **segunda aplicación del instrumento** (*reassessment*) con el mismo procedimiento de análisis del diagnóstico para asegurar comparabilidad antes/después y medir de forma objetiva el impacto de las acciones implementadas.

Síntesis de la ejecución (Acción) La ejecución se organizó de manera incremental, combinando implementación técnica, estandarización de procesos y espacios de socialización. El trabajo se concentró en capacidades de **Entrega Continua, Arquitectura y Gestión Lean y Monitoreo**. Entre los resultados operativos más relevantes se destacan: (i) integración de pruebas automáticas y reportes de cobertura en *GitHub Actions* para cada *pull request*, (ii) consolidación de documentación arquitectónica (diagramas C4), inventario del *stack* y adopción de ADR para decisiones técnicas ([Nygard, 2011](#)), (iii) fortalecimiento de observabilidad y fiabilidad mediante definición inicial de SLIs/SLOs, monitoreo sintético y alertas, alineado con prácticas de operación confiable ([Beyer et al., 2016](#)), y (iv) estandarización de visualización del trabajo y control del flujo mediante documentación mínima, DoR/DoD, tableros Kanban y rituales de revisión periódica ([Anderson, 2010](#); [Poppendieck and Poppendieck, 2003](#)). De forma complementaria, se integró la medición de métricas DORA mediante Apache DevLake para disponer de tableros comparables y trazables en el tiempo ([Apache Software Foundation, 2023](#)).

Síntesis de la evaluación (Reassessment) La evaluación posterior se realizó mediante la replicación del modelo de capacidades, preservando la estructura del cuestionario, la escala de respuesta y la estrategia metodológica de análisis (selección del valor mínimo por ítem) para mantener consistencia con el diagnóstico. Los resultados mostraron un avance generalizado: **15 de 25 capacidades mejoraron, 10 se mantuvieron estables** y no se registraron retrocesos. A nivel agregado, la madurez total aumentó de **2.44 a 3.57** (+1.13), acompañado por un desplazamiento de capacidades hacia categorías superiores y mejoras en el promedio por dimensión. Adicionalmente, se observó crecimiento en **Cultura**, interpretado como un efecto emergente derivado de la estandarización, automatización y visibilidad operativa introducidas por las acciones del plan,

coherente con la literatura que vincula capacidades técnicas y desempeño con prácticas culturales habilitadoras (Kim et al., 2021; Forsgren et al., 2018).

Cierre El capítulo deja evidencia de una iteración completa del ciclo: ejecución con entregables instalados y evaluación cuantitativa comparable frente a la línea base. Los hallazgos permiten fundamentar la continuidad del proceso mediante un siguiente ciclo de mejora, orientado a remover restricciones sistémicas persistentes y a profundizar capacidades habilitadoras, incluyendo la consolidación de monitoreo proactivo y la incorporación progresiva de seguridad en etapas tempranas del ciclo de entrega (*shift-left/DevSecOps*) (Committee, 2021; OWASP Foundation, 2025).

Conclusiones

5.1. Conclusiones

Este trabajo tuvo como propósito evaluar y fortalecer la adopción de prácticas DevOps en el área de producto de *Hello Insight* mediante un ciclo de investigación–acción. En coherencia con los objetivos planteados, se (i) estableció una línea base de madurez a través del modelo de capacidades de Molina (2022) y se complementó con métricas DORA (DevOps Research and Assessment (DORA), 2023; Apache Software Foundation, 2023); (ii) se diseñó y ejecutó un plan de acción de cuatro semanas orientado a cerrar brechas priorizadas; y (iii) se realizó una segunda medición para verificar cambios en las capacidades y dimensiones evaluadas.

En primer lugar, el diagnóstico inicial evidenció un nivel global de madurez de **2.44/5**, clasificado como *Oportunidades de mejora*, con brechas distribuidas en todas las dimensiones del modelo. A partir de la socialización con el equipo y de criterios de impacto y viabilidad, se priorizaron cuatro capacidades de intervención directa, las cuales se tradujeron en actividades técnicas y de proceso que fortalecieron prácticas de entrega continua, estandarización arquitectónica y observabilidad, así como mecanismos de visualización del trabajo. Como resultado, la evaluación mostró un incremento del puntaje global a **3.57/5** (+1.13), junto con un desplazamiento de la mayoría de capacidades hacia categorías superiores (de *Oportunidades de mejora* a *Aceptable* y *Fortalezas*), sin registrar retrocesos. Esta evidencia respalda que una intervención acotada, enfocada y medible puede generar avances significativos en un horizonte corto, especialmente cuando se instala en artefactos verificables (workflows de CI/CD, documentación, tableros, SLIs/SLOs y mecanismos de seguimiento).

En segundo lugar, los hallazgos se alinean con la literatura que describe a DevOps como un enfoque socio-técnico, donde prácticas técnicas (automatización, pruebas, observabilidad) y prácticas organizacionales (colaboración, responsabilidad compartida, aprendizaje) se refuerzan mutuamente (Kim et al., 2021; Forsgren et al., 2018; Bass et al., 2015). En particular, el avance transversal observado —incluyendo mejoras en dimensiones no intervenidas explícitamente— es consistente con la visión de DORA sobre capacidades habilitadoras: la automatización de validaciones, la estandarización del flujo y la visibilidad del trabajo tienden a reducir fricción y ambigüedad, fortaleciendo patrones culturales asociados con cooperación, aprendizaje y enfoque en calidad (Forsgren et al., 2018; DevOps Research and Assessment (DORA), 2023). De manera complementaria, esta lectura también es compatible con enfoques de gestión del cambio, donde la institucionalización de nuevas rutinas ocurre cuando los procesos, sistemas de trabajo y mecanismos de retroalimentación refuerzan consistentemente los comportamientos esperados (Kotter, 1996).

En tercer lugar, las implicaciones prácticas para *Hello Insight* son claras. La organización cuenta ahora con una base operativa mínima para sostener mejoras: (i) un flujo de CI/CD con validaciones y evidencia visible por *pull request*, (ii) documentación centralizada y estandarizada en el *engineering handbook*, (iii) prácticas iniciales de fiabilidad mediante SLIs/SLOs y monitoreo sintético (Beyer et al., 2016), y (iv) reglas y artefactos de visualización del trabajo para mejorar coordinación. En términos de gestión, el principal valor del ciclo no fue únicamente el aumento del puntaje, sino la instalación de mecanismos de mejora continua que permiten iterar de manera más disciplinada, con relación explícita entre diagnóstico, acciones y resultados.

Adicionalmente, se observó que la efectividad tanto del diagnóstico como de la evaluación posterior no depende únicamente del instrumento aplicado, sino también de la selección del público objetivo y de la socialización continua de resultados. La participación de personas cercanas al proceso y con visibilidad sobre las acciones ejecutadas permitió representar con mayor fidelidad la realidad del equipo. Asimismo, la comunicación periódica de resultados facilitó la apropiación del ejercicio y mantuvo el proceso conectado con la operación diaria, incrementando la probabilidad de adopción sostenida de las prácticas definidas.

Desde la perspectiva metodológica, el modelo de capacidades de Molina y las recomendaciones incluidas en su artefacto fueron determinantes para estructurar el proyecto, al proveer una base sistemática para traducir un diagnóstico de madurez en iniciativas concretas de mejora. La clasificación por dimensiones y capacidades permitió identificar brechas prioritarias y, a partir de dichas recomendaciones, derivar actividades ajustadas al contexto y necesidades del área de producto. Este enfoque evitó que el plan de acción se limitara a una lista ad hoc de tareas, y favoreció la construcción de un conjunto de intervenciones justificadas y alineadas con buenas prácticas DevOps, respaldadas por evidencia verificable.

Finalmente, el trabajo deja un resultado metodológico replicable: combinar un instrumento de capacidades (Molina, 2022) con métricas DORA y un plan de acción corto permite diagnosticar, intervenir y medir de manera consistente, generando evidencia útil tanto para decisiones de ingeniería como para aprendizaje organizacional en un contexto real (Adolph et al., 2020). En conjunto, los resultados obtenidos establecen una base para sostener la mejora continua y orientar un siguiente ciclo focalizado en remover restricciones sistémicas, consolidar capacidades habilitadoras y mantener la evolución de la madurez DevOps en el área de producto.

5.2. Amenazas a la validez

Validez de constructo. La medición de capacidades se basa en un cuestionario y, por tanto, depende de la interpretación de los participantes y de su exposición a las prácticas implementadas. Aunque se mantuvieron las mismas definiciones, escala y reglas de análisis (*valor mínimo*) para asegurar comparabilidad, persiste el riesgo de variabilidad en la interpretación de las preguntas o en el entendimiento del alcance de cada práctica (Molina, 2022). Asimismo, la mejora observada en dimensiones no intervenidas puede reflejar cambios reales, pero también cambios en percepción derivados de mayor visibilidad de procesos y artefactos.

Validez interna. Este estudio se desarrolló en un entorno real y no controlado; por tanto, no es posible atribuir causalidad exclusiva del cambio observado a una única intervención. Factores externos (cambios de personal, prioridades de negocio, carga de trabajo, incidentes operativos, o iniciativas paralelas) pudieron influir en los resultados. Para mitigar esta amenaza, se priorizó la trazabilidad entre actividades ejecutadas y evidencias verificables (repositorios, documentación y flujos), y se enfatizó una interpretación prudente del efecto cultural como relación plausible y consistente con literatura, no como causalidad determinista (Forsgren et al., 2018; Kim et al., 2021).

Validez externa. Los resultados corresponden al contexto específico de *Hello Insight* (estructura del equipo, stack tecnológico, prácticas previas y restricciones operativas). Si bien el enfoque metodológico es transferible, el tamaño del equipo, el nivel inicial de madurez y la infraestructura disponible pueden diferir en otras organizaciones; por tanto, la magnitud del efecto y la velocidad de cambio pueden no generalizarse directamente (Bass et al., 2015; Jayakody and Wijayanayake, 2024).

Validez de conclusión. El periodo de intervención (cuatro semanas) limita la capacidad de observar efectos de largo plazo, especialmente en métricas de desempeño que pueden requerir ventanas mayores para estabilizarse (p. ej., MTTR o *lead time*). Adicionalmente, el uso del valor mínimo por ítem, aunque útil para identificar brechas críticas, puede subestimar avances parciales o heterogeneidad positiva dentro del equipo, afectando la sensibilidad del indicador.

5.3. Trabajos futuros

Como continuidad de este trabajo, se propone apropiarse el modelo de capacidades de Molina como práctica recurrente de evaluación y seguimiento en el equipo de desarrollo de *Hello Insight*. El trabajo futuro prioritario consiste en operacionalizar el instrumento mediante ciclos con una cadencia definida (p. ej., bimestral o trimestral), de forma que la organización sostenga un proceso sistemático para identificar brechas, priorizarlas e intervenirlas, y así evitar que las mejoras queden como iniciativas aisladas.

En el primer ciclo posterior, se recomienda concentrar la intervención en capacidades con efecto transversal sobre el desempeño del sistema. En particular, resulta pertinente abordar **Q12** (visibilidad del flujo durante todo el ciclo) y **Q15** (proceso liviano de aprobación de cambios) para mejorar la trazabilidad de extremo a extremo (idea→producción), reducir cuellos de botella y disminuir esperas asociadas a aprobaciones, procurando que los acuerdos de gobernanza se mantengan livianos y no introduzcan fricción innecesaria. De forma complementaria, se sugiere profundizar **Q17** (monitorización proactiva) ampliando SLIs/SLOs a más servicios y formalizando rutinas operativas de respuesta a incidentes (roles, runbooks, postmortems), incorporando presupuestos de error como insumo para priorizar trabajo técnico y balancear confiabilidad con nuevas entregas (Beyer et al., 2016). Asimismo, un trabajo futuro relevante es evolucionar hacia prácticas DevSecOps operacionalizando **Q7** (shift-left de seguridad), integrando análisis estático, escaneo de dependencias, *secret*

scanning y controles automatizados en el *pipeline* alineados con modelos de madurez (OWASP Foundation, 2025).

Adicionalmente, se plantea como línea de trabajo futuro robustecer las métricas DORA mediante la calibración de los criterios operativos de cada indicador y mejoras en la instrumentación y calidad de datos para asegurar comparabilidad entre periodos. Con esta base, se recomienda extender el seguimiento por al menos dos trimestres y evaluar el efecto de las intervenciones comparando periodos pre/post por ciclo, manteniendo trazabilidad entre *actividad* → *capacidad intervenida* → *métrica esperada*. Esto permitiría identificar qué acciones generan mayor efecto, cuáles presentan impacto diferido y cuáles requieren ajustes para sostener mejoras.

Finalmente, se propone como trabajo futuro complementar los futuros ciclos del modelo con un plan de fortalecimiento y/o profundización de capacidades técnicas del equipo, orientado a cerrar brechas de conocimiento que limitan la ejecución consistente de las prácticas DevOps. En la medida en que el equipo eleve sus competencias, se facilita la apropiación del framework, se reduce la dependencia de esfuerzos individuales y se incrementa la probabilidad de que las mejoras se mantengan y se profundicen entre iteraciones.

5.4. Lecciones aprendidas

El desarrollo del proyecto no solo produjo resultados sobre la adopción de prácticas DevOps en *Hello Insight*, sino también aprendizajes derivados del proceso para obtenerlos. Al tratarse de un ciclo de investigación–acción, los hallazgos se construyeron mediante una iteración de diagnóstico, planificación, acción y evaluación, lo cual permitió observar cómo decisiones metodológicas, restricciones del contexto y dinámicas del equipo condicionan tanto la medición como el cambio.

Un aprendizaje central fue que un modelo de capacidades aporta valor cuando se convierte en un mecanismo de toma de decisiones. El instrumento no debe entenderse como un “puntaje final” ni utilizarse como un mecanismo de sanción, sino como una guía para iniciar y estructurar el abordaje del problema: sus preguntas y recomendaciones facilitaron conversaciones informadas sobre el estado real de las prácticas, permitieron explicitar brechas y ayudaron a alinear al equipo en torno a prioridades compartidas. Esto se evidenció al traducir los resultados en un plan acotado de acciones con responsables y evidencias verificables, reduciendo el riesgo de iniciativas reactivas o dispersas.

Asimismo, la experiencia reforzó la importancia de diseñar intervenciones realistas bajo restricciones. Un periodo de intervención corto obliga a seleccionar pocas capacidades con impacto transversal y a privilegiar acciones que puedan institucionalizarse. En este sentido, la efectividad del ciclo dependió más de la claridad del foco, la disciplina de ejecución y la trazabilidad de evidencias que de la cantidad de iniciativas emprendidas.

En el plano de la sostenibilidad, se aprendió que los cambios técnicos y de proceso deben dejar huella en artefactos del sistema de trabajo. Automatizaciones, reglas de revisión, documentación mínima y tableros de seguimiento son valiosos en tanto reducen la dependencia de “lo que solo algunas personas saben” y facilitan que el cambio se mantenga a pesar de rotación, carga operativa o prioridades competitivas. Esto llevó a privilegiar prácticas integradas al flujo diario, en lugar de

acuerdos informales difíciles de sostener.

En relación con la validez y el alcance, se reconoce que, por tratarse de una intervención de corta duración, no todos los efectos pueden observarse de inmediato y algunos pueden aparecer con el tiempo. Asimismo, parte de la medición de capacidades puede estar influenciada por la percepción de los participantes, por lo que resulta importante complementar la evaluación con evidencia verificable (artefactos, *logs*, tableros) y repetir mediciones para reducir variabilidad. Adicionalmente, la comparabilidad de las métricas DORA depende de la consistencia de las definiciones y de la cobertura de las fuentes de datos, lo cual exige esfuerzos continuos de instrumentación y depuración.

Estas lecciones orientan los trabajos futuros, en la medida en que sugieren adoptar el framework como una rutina: repetir ciclos periódicos de evaluación y acción, mantener evidencia verificable de lo implementado y asegurar trazabilidad entre brechas identificadas, actividades ejecutadas y resultados observados. Con ello, las mejoras dejan de depender de iniciativas puntuales y se convierten en una práctica estable del equipo.

5.5. Cierre

En síntesis, la investigación demostró que *Hello Insight* podía avanzar de un estado inicial de madurez hacia un nivel *Acceptable* mediante una intervención breve pero estructurada, trazable y apoyada en evidencia. El principal aporte del trabajo no fue únicamente el incremento del puntaje, sino la instalación de una base operativa y cultural para sostener la mejora continua: prácticas automatizadas, documentación mínima, observabilidad inicial y mecanismos de coordinación que permiten iterar con mayor disciplina y menor fricción.

Capacidades con oportunidades de mejora y plan de acción sugerido por el artefacto de Molina

Capacidad	Plan de acción
Aumentar el proceso de despliegue	Iniciar la adopción de herramientas que automaticen la compilación y generación del producto final, ejemplo adopción de Maven, Gradle para ambientes Java, Nuget para C#, adicionalmente integrar procesos de aseguramiento de calidad, como son pruebas manuales (unitarias y de aceptación). Por último, definir roles y responsabilidades en común acuerdo entre desarrollo y operaciones de quienes son los responsables a la hora de hacer los despliegues.
Utilizar arquitecturas de bajo acoplamiento	La compañía debe iniciar un plan de organización y estandarización del stack tecnológico existente con el fin de conocer qué elementos pueden ser compartidos y reutilizables en diferentes condiciones, de modo que en un futuro se pueda eliminar la necesidad de hacer configuraciones y actualizaciones manuales de piezas individuales de hardware. También identificar y depurar los entornos particulares o exclusivos que no pueden reproducirse automáticamente.
Tener un proceso liviano para la aprobación de cambios	Se debe iniciar la adopción de procedimientos y herramientas que permitan gestionar y visualizar los cambios de un proyecto. Lo principal es dar orden y visibilidad a dichas solicitudes para que puedan ser analizadas y aprobadas por los mecanismos y metodologías que use la compañía (revisión por pares, revisión interequipo, evaluación 360).
Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	La compañía debe adoptar la cultura de generar documentación mínima necesaria para obtener fácilmente el estado actual de un proyecto.

Capacidad	Plan de acción
Implementar gestión de los datos de prueba	Se debe adoptar un proceso colaborativo entre desarrolladores, infraestructura y calidad para definir el criterio, alcance y los datos que se utilizarán durante las pruebas. Se recomienda priorizar las pruebas unitarias, ya que son la base de la pirámide de pruebas, y progresivamente escalar en los demás tipos de pruebas (aceptación, integración, etc.). Además, asegurar que existan datos de prueba adecuados, consistentes y realistas para garantizar la correcta ejecución de los casos. Este proceso debe iniciarse desde las fases tempranas de los proyectos, aplicando prácticas como <i>Test-Driven Development</i> .
Implementar entrega continua de software	Se debe iniciar con la adopción de tareas automáticas partiendo desde un sistema de control de versiones centralizado, en el que se ejecuten pruebas unitarias y análisis de calidad de código estático.
Recolectar e implementar las recomendaciones (feedback) de los clientes	El feedback de los clientes debe incorporarse en todo el ciclo de vida del producto, a través de la recopilación activa y periódica de la opinión de los usuarios finales y la definición de métricas de satisfacción.
Hacer visible el flujo de trabajo durante todo el ciclo	La compañía debe asegurar que el equipo conozca y participe desde el inicio en la definición del flujo de trabajo, desde la solicitud del cliente hasta la entrega del producto. Esto incluye definir bloques de trabajo, personal involucrado, actividades, tiempos estimados y apoyarse en herramientas de visualización que permitan consultar el estado de cada etapa.
Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	El monitoreo debe ir más allá de las estadísticas de uso de recursos físicos o del servidor. Se deben adoptar herramientas que midan la salud real de las aplicaciones, pues no basta que el servidor esté activo si el usuario final no puede utilizar la aplicación.
Implementar una cultura tipo generativa	Para evolucionar hacia este modelo, la compañía debe aceptar el paradigma de constante evolución y adaptarse a los cambios del entorno, priorizando la innovación, la meritocracia y el cumplimiento de objetivos. Se debe dejar atrás estructuras jerárquicas rígidas para impulsar la colaboración y la mejora continua.

Capacidad	Plan de acción
Apoyar y facilitar la colaboración entre equipos	Propiciar equipos multidisciplinarios con representantes de cada área funcional (desarrollo, análisis, calidad, infraestructura, seguridad, etc.) y crear canales de comunicación abiertos y colaborativos. De esta forma, la toma de decisiones será compartida desde el inicio, y la responsabilidad de compilar, implementar y mantener un producto recaerá en todos. La automatización debe apoyarse con herramientas adecuadas que refuercen la colaboración.
Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos	La compañía debe permitir flexibilidad en el stack tecnológico corporativo para adaptarse a las particularidades de cada proyecto, brindando al equipo las herramientas y recursos adecuados. No todas las necesidades pueden resolverse con una única herramienta universal.
Apoyar el liderazgo transformacional	Iniciar un proceso para transformar a los líderes hacia un enfoque de liderazgo tecnológico, que fomente la confianza, el apoyo, el trabajo colaborativo y la innovación. Un buen liderazgo influye directamente en la capacidad del equipo para entregar código de calidad, diseñar arquitecturas escalables y aplicar principios LEAN en la gestión del trabajo.

Tabla A.1: Capacidades con oportunidades de mejora y su plan de acción

Definición detallada de las actividades del plan de acción definitivo

B.0.0.1. Entrega Continua — Gestión de datos de prueba

Objetivo: Asegurar la disponibilidad y calidad de los datos de prueba, priorizando pruebas unitarias y escalando progresivamente a otros niveles.

1. **Documentar estado actual de las pruebas: tipo de pruebas, frameworks y cobertura**
 - **DoD:** Documento en `/docs/testing-state.md`
 - **Esfuerzo:** 4 h
 - **Semana:** 1
2. **Definir alcance de pruebas unitarias por módulo (backend, frontend, analytics)**
 - **DoD:** Documento en `/docs/testing-scope.md`
 - **Esfuerzo:** 4 h
 - **Semana:** 1
3. **Definir proceso para garantizar disponibilidad de los datos de prueba**
 - **DoD:** Documento en `/docs/testing-requirements.md`
 - **Esfuerzo:** 4 h
 - **Semana:** 2
4. **Ejecutar pruebas en CI + publicar cobertura**
 - **DoD:** Workflow activo, badge en README, reporte en PR
 - **Esfuerzo:** 6 h
 - **Semana:** 3
5. **Garantizar umbral mínimo de cobertura (gate)**
 - **DoD:** Pipeline falla si cobertura $< 60\%$

- **Esfuerzo:** 4 h
- **Semana:** 3

6. Sesión de socialización (pruebas + datos)

- **DoD:** Acta con acuerdos
- **Esfuerzo:** 2 h
- **Semana:** 4

B.0.0.2. Arquitectura — Bajo acoplamiento

Objetivo: Estandarizar el stack tecnológico para facilitar escalabilidad y mantenibilidad.

1. Generar Diagramas C4 de Contexto y Contenedores

- **DoD:** Diagrama en /docs/c4.md
- **Esfuerzo:** 3 h
- **Semana:** 1

2. Inventario de stack con estado y riesgo

- **DoD:** Tabla en /docs/stack-inventory.md
- **Esfuerzo:** 4 h
- **Semana:** 1

3. Proceso ligero para elección de nuevas herramientas

- **DoD:** Checklist y plantilla ADR
- **Esfuerzo:** 4 h
- **Semana:** 2

4. Sesión de socialización (documentos)

- **DoD:** Acta con acuerdos
- **Esfuerzo:** 2 h
- **Semana:** 4

B.0.0.3. Gestión Lean y Monitorización — Aplicaciones e Infraestructura

Objetivo: Medir la salud de aplicaciones e infraestructura más allá de métricas de hardware, con foco en experiencia de usuario y KPIs de negocio.

1. Documentar herramientas actuales de monitorización y alertas

- **DoD:** Documento en docs/monitors.md
- **Esfuerzo:** 6 h
- **Semana:** 1

2. Definir estrategias de revisión periódicas y visibilidad de las herramientas

- **DoD:** Documento en /docs/monitors.md
- **Esfuerzo:** 4 h
- **Semana:** 2

3. Pipelines de despliegue para métricas DORA (DF, LT, CFR, MTTR)

- **DoD:** Workflow con export de métricas
- **Esfuerzo:** 8 h
- **Semana:** 2

4. Dashboard DORA

- **DoD:** Visual con 4 métricas principales
- **Esfuerzo:** 6 h
- **Semana:** 3

5. Monitoreo sintético de login

- **DoD:** Script de journey con alerta
- **Esfuerzo:** 4 h
- **Semana:** 3

6. Definir 1 SLI/SLO y alertas

- **DoD:** Documento + alerta activa en canal
- **Esfuerzo:** 4 h
- **Semana:** 3

7. Sesión de socialización (pipelines y accesos)

- **DoD:** Acta con acuerdos
- **Esfuerzo:** 2 h
- **Semana:** 4

B.0.0.4. Gestión Lean y Monitorización — Visualizar el trabajo

Objetivo: Generar documentación mínima y tableros visuales que permitan conocer el estado de los proyectos y comunicar la calidad al equipo.

1. Estándar de documentación mínima

- **DoD:** /docs/README-template.md con ejemplo
- **Esfuerzo:** 4 h
- **Semana:** 1

2. Proceso de calidad (diseño/requisitos/UI)

- **DoD:** /docs/quality-checklist.md aplicado en un PR
- **Esfuerzo:** 6 h
- **Semana:** 1

3. Tablero Kanban estándar

- **DoD:** Columnas definidas, WIP limits, etiquetas
- **Esfuerzo:** 4 h
- **Semana:** 1

4. Visualización central (Notion/Git)

- **DoD:** Página índice de estado de proyectos
- **Esfuerzo:** 4 h
- **Semana:** 2

5. Guías rápidas de mantenimiento de documentación

- **DoD:** Proceso definido para mantener docs
- **Esfuerzo:** 6 h
- **Semana:** 3

6. Ritual de revisión quincenal

- **DoD:** Evento recurrente + minuta 1 pág.
- **Esfuerzo:** 2 h
- **Semana:** 4

Formato para ADR en Hello Insight

Instrucciones

Esta encuesta se diligencia al proponer una nueva tecnología, servicio o herramienta, o al realizar un cambio arquitectónico relevante. El objetivo es recopilar la información mínima necesaria para redactar y aprobar un ADR.

Identificación

- **Título de la decisión:** _____
- **Fecha:** _____
- **Autor(es):** _____
- **Equipo/Área:** _____
- **Repositorio/Servicio impactado:** _____
- **Estado del ADR:** Propuesto Aprobado Rechazado Deprecado

Cuestionario

1. Nombre y categoría

- ¿Cómo se llama la tecnología?
- ¿A qué capa pertenece? (Frontend, APIs, Datos, Infraestructura, Seguridad, Herramientas, Servicios Externos)

2. Propósito

- ¿Qué problema resuelve?
- ¿Por qué se eligió sobre otras alternativas?

3. Uso en Hello Insight

- ¿Cómo se utiliza en nuestros sistemas o flujos de trabajo?
- ¿Qué aplicaciones o equipos dependen de ella?

4. Dependencias e integraciones

- ¿Se conecta con otras herramientas, APIs o servicios?
- ¿Requiere alguna versión específica, plugin o librería?

5. Propiedad y acceso

- ¿Quién la mantiene (equipo o rol)?
- ¿Dónde se almacenan las credenciales o configuraciones (por ejemplo, 1Password, GitHub Secrets)?

6. Ciclo de vida

- ¿Qué versión estamos usando actualmente?
- ¿Existe un plan de actualización, migración o desuso (deprecación) con fechas estimadas?

7. Seguridad y cumplimiento

- ¿Hay controles de acceso, cifrado o requisitos de cumplimiento (SOC2, GDPR, etc.)?
- ¿Existen consideraciones sobre residencia de datos o manejo de PII/datos personales?

8. Documentación

- ¿Dónde pueden los ingenieros encontrar guías de instalación/configuración?
- ¿Hay *runbooks* internos o ADRs enlazados?

Script de procesamiento de respuestas

Este anexo presenta el script utilizado para mapear las respuestas textuales del cuestionario a valores numéricos (*RID*) y aplicar la estrategia metodológica de seleccionar, por ítem, el valor de menor denominación con el fin de representar la brecha más crítica del equipo. En el entorno original, el procesamiento se ejecutó en Google Colab; por claridad y reproducibilidad, el listado a continuación se presenta en formato Python estándar (sin comandos específicos del notebook). El tratamiento de respuestas *No sabe/No responde* y *No aplica* se realiza ignorando valores nulos en el cálculo.

```
1 # -*- coding: utf-8 -*-
2
3 import pandas as pd
4 from fuzzywuzzy import process
5
6 # Dependencias (instalar en el entorno):
7 # pip install fuzzywuzzy python-Levenshtein
8
9 # Rutas de archivos (ajustar según el entorno)
10 RESPUESTAS_CSV = "respuestas.csv"
11 CATALOGO_CSV = "bd_respuestas_accion.csv"
12
13 # Cargar datos
14 r_df = pd.read_csv(RESPUESTAS_CSV, delimiter=";")
15 p_df = pd.read_csv(CATALOGO_CSV)
16
17 # Seleccionar columnas de preguntas (según estructura original)
18 # Nota: este rango depende del layout del CSV exportado desde Forms
19 response_columns = r_df.columns[7:32]
20 r_df_responses = r_df[["Id"] + response_columns.tolist()]
21
22 # DataFrame de salida: Id + columnas RID
23 r_df_numerical = r_df_responses[["Id"]].copy()
24
25 # Mapear respuestas a RID por pregunta usando fuzzy matching
26 for col in response_columns:
27     rid_col = col + "_RID"
28     r_df_numerical[rid_col] = None
29
30 # Respuestas posibles en el catalogo para la pregunta
31 valid_answers = p_df[p_df["Question"] == col]["Respuesta"].dropna().unique()
32
33 for index, response in r_df_responses[col].items():
```

```
34     if pd.isna(response):
35         continue
36
37     best_match = process.extractOne(response, valid_answers, score_cutoff=95)
38     if best_match:
39         matched_response = best_match[0]
40         rid_value = p_df[
41             (p_df["Question"] == col) &
42             (p_df["Respuesta"] == matched_response)
43         ]["RID"].iloc[0]
44         r_df_numerical.loc[index, rid_col] = rid_value
45
46     # Calculo del minimo por pregunta (estrategia del valor mas bajo)
47     rid_columns = [c for c in r_df_numerical.columns if c.endswith("_RID")]
48     rid_numeric = r_df_numerical[rid_columns].apply(pd.to_numeric, errors="coerce")
49     min_responses = rid_numeric.min(axis=0, skipna=True)
50
51     # Resultados principales (pueden exportarse para trazabilidad)
52     r_df_numerical.to_csv("respuestas_mapeadas_rid.csv", index=False)
53     min_responses.to_csv("minimo_por_pregunta.csv", header=["min_value"])
```

Bibliografía

- Adolph, S., Kruchten, P., and Melo, W. (2020). *Action Research in Software Engineering: Theory and Applications*. Springer.
- Amaro, R., Pereira, R., and Mira da Silva, M. (2023). Capabilities and practices in devops: A multivocal literature review. *IEEE Transactions on Software Engineering*, 49(2):883–901. Available online: <https://doi.org/10.1109/TSE.2022.3166626>.
- Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- Apache Software Foundation (2023). Apache devlake: Dora. <https://devlake.apache.org/docs/DORA/>. Accessed: 2025-02-21.
- Atlassian (n.d.). Calms framework.
- Bass, L., Weber, I. M., and Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley.
- Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (2016). *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media.
- CMMI Institute (2025). CMMI Institute - Improving Process Maturity. Accessed: 2025-03-18.
- Cockburn, A. (2001). Agile software development. In *Encyclopedia of Software Engineering*. Wiley.
- Committee, S. . S. E. S. (2021). Ieee standard for devops:building reliable and secure systems including application build, package, and deployment. *IEEE Std 2675-2021*, pages 1–91.
- Crispin, L. and Gregory, J. (2009). *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison-Wesley Professional, Boston. Referenced from pages 344–346 (Tester Activities in Release or Theme Planning).
- DevOps Research and Assessment (DORA) (2023). Accelerate: Informe del estado de devops 2023. https://dora.dev/research/2023/dora-report/2023-dora-accelerate-state-of-devops-report_es.pdf. Consultado el 3 de septiembre de 2025.
- Forsgren, N., Humble, J., and Kim, G. (2018). *Accelerate: Building and Scaling High Performing Technology Organizations*.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Addison-Wesley.

- Jayakody, J. A. and Wijayanayake, W. M. (2024). Devops maturity; a systematic literature review. In *Proceedings - International Research Conference on Smart Computing and Systems Engineering (SCSE)*.
- Kato, D., Shimizu, A., and Ishikawa, H. (2022). Quality classification for testing work in devops. pages 156–162.
- Kim, G., Humble, J., Debois, P., Willis, J., and Forsgren, N. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press.
- Kotter, J. P. (1996). *Leading Change*. Harvard Business School Press, Boston, MA.
- Molina, J. (2022). Evaluación de la conveniencia de adoptar devops como estrategia para la entrega continua de productos de software en la alianza bioversity - ciat sede las américas, colombia. Trabajo de grado para optar al título de magister en ingeniería de software, Pontificia Universidad Javeriana - Cali, Santiago de Cali, Colombia. Disponible en: <http://hdl.handle.net/11522/2164>.
- Nygaard, M. (2011). Documenting architecture decisions. <https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>. Accessed: 2025-02-21.
- OWASP Foundation (2025). DevSecOps Maturity Model (DSOMM). Accessed: 2025-03-18.
- Poppendieck, M. and Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.
- Pérez-Mergarejo, E., Pérez-Vergara, I., and Rodríguez-Ruíz, Y. (2014). Modelos de madurez y su idoneidad para aplicar en pequeñas y medianas empresas. *Ingeniería Industrial*, XXXV(2):146–158.
- Wilson, C. (2010). Chapter 4 - brainstorming. In Wilson, C., editor, *User Experience Re-Mastered*, pages 107–134. Morgan Kaufmann, Boston.