



Escuela de Administración de Tecnologías de la Información

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa manufacturera mediante la metodología DevOps CI/CD

Trabajo Final de Graduación para optar al grado de Licenciatura en Administración de Tecnologías de Información

Modalidad Proyecto de Graduación

Elaborado por Daniel Eduardo Obaldía Solís

Prof. Tutor: Lic. Michael Sánchez Soto

Cartago, Costa Rica

I semestre, 2025

 CC BY-NC-SA 4.0

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa manufacturera mediante la metodología DevOps CI/CD © 2025 by Daniel Eduardo Obaldía Solís is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Hoja de Aprobación

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN
GRADO ACADÉMICO: LICENCIATURA

Los miembros del Tribunal Examinador de la Escuela de Administración de Tecnologías de Información recomendamos que el siguiente informe del Trabajo Final de Graduación del estudiante Daniel Eduardo Obaldía Solís sea aceptado como requisito parcial para obtener el grado académico de Licenciatura en Administración de Tecnologías de Información.

Lic. Michael Sánchez Soto. Profesor tutor



Lic. Alberto Solano Solano. Lector externo

Lic. Néstor Morales Rodríguez. Lector académico

M. Sc.. Yarima Sandoval Sánchez. Coordinadora de Trabajo Final de Graduación

Dedicatoria

A mi mamá y mi abuela, quienes han sido mi apoyo incondicional y motivación.

A mi hermano, quien me impulsa a siempre ser mejor en lo que hago.

A mi tía, que siempre es una luz cuando todo se oscurece.

A mis amigos, por estar siempre ahí para hacerme reír en los momentos difíciles.

Resumen

Este Trabajo Final de Graduación tiene como objetivo desarrollar una propuesta de estandarización del marco de trabajo para la gestión de proyectos del equipo de Ingeniería de Inteligencia Artificial en una empresa manufacturera multinacional. Para ello, se aplica la metodología DevOps con un enfoque en la aplicación de CI/CD, con el fin de mejorar los procesos de planificación, documentación, análisis, desarrollo, pruebas y despliegue de proyectos, específicamente en la Unidad de Cuidado Familiar, durante el primer semestre del año 2025.

La investigación sigue un enfoque mixto y un diseño de investigación-acción, dividido en cuatro fases: diagnóstico de la situación actual del equipo, diseño del marco de trabajo, comparación entre el estado actual y la propuesta, y la implementación de una prueba de concepto. Como parte del proceso metodológico, se emplearon entrevistas, encuestas, revisión documental, observación directa y modelado de procesos BPMN.

Los hallazgos de este proyecto permiten evidenciar deficiencias en la estandarización de la gestión de proyectos, escasa documentación, limitaciones técnicas y el incumplimiento de criterios internos de calidad. En respuesta, se diseña un marco de trabajo que incorpora herramientas como GitHub Actions, SonarQube, Databricks Asset Bundles, entre otros procedimientos que permiten una integración continua y despliegue automatizado dentro del ciclo DevOps.

La prueba de concepto realizada evidencia mejoras en la calidad del código, reducción de errores, incremento en la frecuencia de despliegue y una mayor trazabilidad. Además, el marco de trabajo establece ser un puente de colaboración más efectivo, mediante la viabilidad técnica y operativa del marco propuesto.

Palabras clave: DevOps, CI/CD, inteligencia artificial, AI Engineer, Databricks, AI Readiness Criteria, mejora continua.

Abstract

This Final Graduation Project aims to develop a standardized framework for project management within the Artificial Intelligence Engineering team of a multinational manufacturing company. The proposal applies the DevOps methodology with a focus on CI/CD implementation, aiming to enhance the processes of planning, documentation, analysis, development, testing, and deployment of AI-related projects, specifically within the Family Care Business Unit, during the first semester of 2025.

The research follows a mixed-method approach and an action-research design, divided into four phases: diagnosis of the current situation, design of the framework, comparison between the current and proposed states, and the implementation of a proof of concept. To achieve this, various data collection techniques were used, including interviews, surveys, document reviews, direct observation, and BPMN-based process modeling.

The project findings reveal deficiencies in project standardization, limited documentation, technical constraints, and a lack of alignment with internal quality standards. As a solution, the proposed framework integrates tools such as GitHub Actions, SonarQube, and Databricks Asset Bundles to enable continuous integration and automated deployment within the DevOps cycle.

The implemented proof of concept demonstrated improvements in code quality, error reduction, increased deployment frequency, and enhanced traceability. Moreover, the framework facilitates more effective collaboration between the AI Engineering team and data facilitators, confirming the technical and operational viability of the proposed solution.

Keywords: DevOps, CI/CD, artificial intelligence, AI Engineer, Databricks, AI Readiness Criteria, continuous improvement.

Tabla de Contenidos

1. Introducción.....	1
1.1. Descripción general.....	1
1.2. Antecedentes.....	2
1.2.1. Descripción de la organización.....	2
1.2.2. Trabajos similares realizados dentro y fuera de la organización.....	4
1.3. Planteamiento del problema.....	6
1.3.1. Situación problemática.....	6
1.3.2. Justificación del proyecto.....	9
1.3.3. Beneficios esperados o aportes del Trabajo Final de Graduación.....	10
1.4. Objetivos del Trabajo Final de Graduación.....	12
1.4.1. Objetivo general.....	12
1.4.2. Objetivos específicos.....	12
1.5. Alcance.....	13
1.6. Supuestos.....	15
1.7. Entregables.....	15
1.8. Limitaciones.....	16
2. Marco Conceptual.....	17
2.1. Conceptos prácticos y teóricos.....	18
2.1.1. Gestión de la calidad.....	18
2.1.2. Metodologías ágiles.....	18
2.1.3. DevOps.....	19
2.1.4. Continuous Integration, Continuous Delivery y Continuous Deployment.....	20
2.1.5. Gestión de proyectos de IA.....	21
2.2. Métodos y procedimientos.....	22
2.2.1. DevOps Lifecycle.....	22
2.2.2. CI/CD Pipeline.....	24
2.2.3. Pruebas automatizadas.....	25
2.2.4. Code Review.....	26
2.2.5. Evaluación de madurez de procesos con prácticas DevOps.....	26
2.3. Herramientas, prácticas y tecnologías.....	28
2.3.1. Herramientas de CI/CD.....	28
2.3.2. Git Flow.....	29
2.3.3. Herramientas de seguimiento.....	30
2.3.4. Herramientas de documentación.....	30
3. Marco Metodológico.....	32
3.1. Tipo de investigación.....	32
3.2. Enfoque y diseño de la investigación.....	32
3.3. Fuentes de datos e información.....	34

3.3.1.	Fuentes primarias	35
3.3.2.	Fuentes secundarias.....	35
3.4.	Sujetos de investigación.....	37
3.5.	Variables o categorías de la investigación.....	39
3.6.	Técnicas e instrumentos de recolección de datos	44
3.6.1.	Revisión documental.....	44
3.6.2.	Entrevista Semiestructurada	45
3.6.3.	Encuesta.....	45
3.6.4.	Observación directa.....	45
3.6.5.	Modelado de procesos BPMN.....	45
3.6.6.	Prueba de concepto.....	46
3.7.	Procedimiento metodológico de la investigación.....	46
3.7.1.	Diagrama propuesto para las fases del procedimiento metodológico	46
3.8.	Operacionalización de las variables.....	49
3.9.	Tabla resumen del procedimiento metodológico de la investigación	51
4.	<i>Análisis de Resultados</i>	53
4.1.	Fase 1: Análisis de la situación actual.....	53
4.1.1.	Resultados de entrevistas semiestructuradas.....	53
4.1.2.	Diagrama Ishikawa de la situación actual del equipo de Ingeniería de Inteligencia Artificial	61
4.1.3.	Revisión documental interna	62
4.1.4.	Análisis comparativo entre las entrevistas realizadas y la documentación interna	71
4.1.5.	Observación directa.....	75
4.1.6.	Revisión documental de marcos de referencia	76
5.	<i>Propuesta de Solución</i>	87
5.1.	Fase 2: Diseño del marco de trabajo.....	87
5.1.1.	Herramientas y procedimientos propuestos según las necesidades del equipo.....	88
5.1.2.	Especificaciones administrativas, técnicas y operativas del marco DevOps CI/CD.....	94
5.2.	Fase 3: Comparación de la situación actual versus Propuesta de marco de trabajo para la Gestión de los proyectos de Ingeniería de Inteligencia Artificial.....	106
5.2.1.	Procesos modificados e integrados al pipeline de desarrollo	106
5.3.	Fase 4: Resultados de la prueba de concepto	108
5.3.1.	Evaluación de éxito de la prueba según el marco de trabajo propuesto	108
5.3.2.	Contexto del proyecto piloto en la Unidad de Cuidado Familiar.....	108
5.3.3.	Implementación de especificaciones y procedimientos técnicos DevOps CI/CD	109
5.3.4.	Resultados de la prueba de concepto	121
5.4.	Análisis costo-beneficio	129
5.4.1.	Determinación de los costos totales proyecto	129
5.4.2.	Cálculo del retorno de inversión	130
6.	<i>Conclusiones</i>	137
6.1.	Conclusiones del objetivo 1	137

6.2.	Conclusiones del objetivo 2	139
6.3.	Conclusiones del objetivo 3	140
6.4.	Conclusiones del objetivo 4	141
7.	Recomendaciones	143
7.1.	Recomendaciones generales	143
7.2.	Recomendaciones por objetivo	144
8.	Referencias.....	146
9.	Apéndices	150
9.1.	Apéndice A. Plantilla de minutas	150
9.2.	Apéndice B. Plantilla de gestión de cambios	151
9.3.	Apéndice C. Cronograma del proyecto	152
9.4.	Apéndice D. Plantilla de revisión documental	153
9.5.	Apéndice E. Plantilla de revisión documental interna en la organización	154
9.6.	Apéndice F. Plantilla para entrevista semiestructurada	155
9.7.	Apéndice G. Entrevista N.º 1	156
9.8.	Apéndice H. Entrevista N.º 2	158
9.9.	Apéndice I. Revisión documental interna de un proyecto en operación del equipo de Ingeniería de Inteligencia Artificial.....	160
9.10.	Apéndice J. Revisión documental interna de un proyecto en operación del equipo de Ingeniería de Inteligencia Artificial desarrollado para la Unidad de Cuidado Familiar.	161
9.11.	Apéndice K. Revisión documental interna de las especificaciones del AI Readiness Criteria	162
	Revisión documental interna #3	162
9.12.	Apéndice L. Revisión documental: <i>The DevOps Handbook 2021</i>	163
	Revisión Documental #1	163
9.13.	Apéndice M. Revisión documental: <i>Continuous Integration and Continuous Delivery (2023)</i>	164
	Revisión Documental #1	164
9.14.	Apéndice N. Revisión documental: <i>Accelerate: The Science of Lean Software and DevOps (2018)</i>	165
	Revisión Documental #3	165
9.15.	Apéndice O. Bitácora de reuniones.....	166

9.16. Apéndice P. Diagrama de Modelo de datos de un proyecto de la Unidad de Cuidado Familiar analizado en la documentación interna.....	167
10. Anexos	168
I. Glosario	168
II. Carta de Revisión Filológica.....	173

Índice de Figuras

Figura 1: Organigrama de la organización.....	3
Figura 2: Árbol del problema.....	8
Figura 3: Estructura del Marco Conceptual.....	17
Figura 4: DevOps Lifecycle.....	23
Figura 5: El pipeline de despliegue.....	24
Figura 6: Flujo de pruebas automatizadas.....	25
Figura 7: Lead Time versus Process time.....	27
Figura 8: Git Flow Model.....	29
Figura 9: Diagrama del procedimiento metodológico.....	48
Figura 10: Diagrama Ishikawa de causas raíz de la situación actual del equipo.....	61
Figura 11: Estructura de modelo de soporte general de un proyecto interno en producción.....	64
Figura 12: Gestión del flujo de valor mediante Lead Time.....	79
Figura 13: IT Value Chain.....	83
Figura 14: Fases ALM.....	84
Figura 15: Vista a alto nivel de CI/CD pipelines con bundles.....	86
Figura 16: Flujo de la fase de planificación.....	97
Figura 17: Flujo de la fase de desarrollo y testing.....	100
Figura 18: Flujo de la fase de despliegue y monitoreo.....	103
Figura 19: Fases del desarrollo del pipeline para el marco propuesto.....	104
Figura 20: Flujo de ejecución del Pipeline.....	105
Figura 21: Pruebas unitarias de la prueba de concepto.....	110
Figura 22: Visualización de SonarQube en la prueba de concepto.....	111
Figura 23: Code Review mediante Copilot 1.....	112
Figura 24: Code Review mediante Copilot 2.....	113
Figura 25: Definición de parámetros del pipeline.....	114
Figura 26: Instalación de dependencias y entorno virtual.....	115
Figura 27: Ejecución de pruebas unitarias y generación de reporte de cobertura.....	116
Figura 28: Ejecución de SonarQube.....	116
Figura 29: Despliegue y ejecución mediante Databricks Asset Bundles.....	117
Figura 30: Visualización de los pasos de ejecución del pipeline.....	118
Figura 31: Workflow de desarrollo en Databricks.....	119
Figura 32: Métricas de ejecución de los tasks del pipeline.....	120
Figura 33: Workflow del pipeline en Databricks.....	120
Figura 34: Métrica de tiempo de ejecución mediante código.....	121
Figura 35: Métrica de tiempo de ejecución mediante la creación de jobs.....	122
Figura 36: Ejecución de la tarea Standard Naming con el procedimiento anterior.....	123
Figura 37: Ejecución de la tarea Standard Naming con el procedimiento anterior mediante código.....	123
Figura 38: Ejecución de la tarea Demand File 1 mediante la creación de jobs.....	124
Figura 39: Ejecución de la tarea Demand File 1 mediante código.....	124
Figura 40: Ejecución de la tarea Demand File 2 mediante la creación de jobs.....	125
Figura 41: Ejecución de la tarea Demand File 2 mediante código.....	125

Índice de Tablas

Tabla 1: Fases del proyecto según el alcance	13
<i>Tabla 2: Fuentes primarias</i>	35
<i>Tabla 3: Fuentes secundarias</i>	35
<i>Tabla 4: Sujetos de investigación</i>	37
Tabla 5: Variables de investigación del objetivo 1.....	39
Tabla 6: Variables de investigación objetivo 2.....	41
Tabla 7: Variables de investigación objetivo 3.....	42
Tabla 8: Variables de investigación objetivo 4.....	43
<i>Tabla 9: Descripción de las fases del procedimiento metodológico</i>	46
Tabla 10: Operacionalización de variables.....	49
Tabla 11: Tabla resumen del procedimiento metodológico de la investigación.....	51
Tabla 12: Componentes del AI Readiness Criteria.....	62
Tabla 13: Análisis de la documentación interna de un proyecto en producción de Ingeniería de Inteligencia Artificial.....	67
Tabla 14: Branching Strategy	70
Tabla 15: Análisis comparativo entre entrevistas y la documentación interna	71
Tabla 16: Correlación entre necesidades, herramientas y practicas	88
Tabla 17: Especificaciones del marco de trabajo en la fase de planificación.....	94
Tabla 18: Especificaciones del marco de trabajo en la fase de desarrollo y testing.....	98
Tabla 19: Especificaciones del marco de trabajo en la fase de despliegue y monitoreo	101
Tabla 20: Tabla comparativa de la situación actual versus Procesos modificados	106
Tabla 21: Resultados de las ejecuciones del pipeline.....	127
Tabla 22: Resultados de las ejecuciones según métricas DevOps.....	127
Tabla 23: Determinación de los costos totales del proyecto.....	129
Tabla 24: Costos de la nueva solución de Databricks	130
Tabla 25: Costos de las especificaciones técnicas.....	131

1. Introducción

1.1. Descripción general

En la industria de tecnologías de información, las organizaciones actuales buscan adoptar metodologías ágiles y buenas prácticas para el desarrollo de soluciones tecnológicas. Por este motivo, la implementación de un marco de trabajo que garantice el despliegue eficiente y estable de proyectos sigue siendo un desafío significativo en la gestión de tecnologías de información.

En este contexto, el equipo de Ingeniería de Inteligencia Artificial de una empresa manufacturera enfrenta dificultades en la gestión efectiva de los proyectos, debido a que existen múltiples enfoques para el desarrollo e implementación de soluciones. Esta empresa opera a través de unidades de negocio, responsables de la gestión, comercialización y manufactura de sus principales categorías de productos. En particular, el desarrollo de este estudio se desarrolla en la Unidad de Cuidado Familiar, que depende de los proyectos del equipo de Ingeniería de Inteligencia Artificial para la implementación de soluciones y automatizaciones que mejoren sus procesos de negocio. Sin un desarrollo adecuado, la operatividad de esta unidad de negocio podría verse afectada.

Actualmente, el equipo de Ingeniería de Inteligencia Artificial busca mejorar la calidad en el desarrollo de sus proyectos. Para lograrlo, es necesario abordar las limitaciones que dificultan la integración de un marco de trabajo estandarizado, el cual permita mejorar la colaboración, trazabilidad de soluciones y la adopción de prácticas ágiles alineadas con los estándares de la industria.

Para abordar esta problemática, se desarrolla una propuesta de marco de trabajo estandarizado basado en la metodología DevOps CICD, enfocado en optimizar las diferentes fases del ciclo de vida de los proyectos dentro de la Unidad de Cuidado Familiar de la organización. La propuesta se implementa mediante un enfoque cualitativo de investigación, con un diseño basado en la metodología de investigación-acción, lo que permite establecer las pautas y conceptos fundamentales para la creación de un marco de trabajo competitivo alineado con las mejores prácticas de la industria.

Este trabajo de graduación busca aplicar una metodología práctica que permita abordar directamente la problemática en un contexto real, como lo es la Unidad de Cuidado Familiar. Asimismo, se incluye la realización de una prueba de concepto dentro de esta unidad de negocio, con el propósito de evaluar la alineación del marco de trabajo propuesto con los objetivos específicos en términos de efectividad y viabilidad.

El desarrollo de este proyecto responde ante los lineamientos establecidos por la Escuela de Administración de Tecnologías de Información del Instituto Tecnológico de Costa Rica. Con ello se busca optar por el grado de Licenciatura en el I semestre del 2025.

1.2. Antecedentes

1.2.1. Descripción de la organización

La organización corresponde a una empresa de bienes de consumo a nivel internacional, la cual cuenta con más de 60 marcas dentro de la industria. La misión de esta compañía es gestionar sus productos de la mejor forma posible, aplicando los mejores procedimientos en la industria de bienes de consumo y manufactura, para mejorar la satisfacción de los clientes mediante productos de una alta calidad. Actualmente, la empresa cuenta con un impacto alrededor de 5000 personas, en 180 países.

La empresa cuenta con una sede en Costa Rica, la cual opera un centro de servicios que respalda distintas áreas y unidades de negocio a nivel mundial. Este centro de negocio se especializa tanto en las tecnologías de información como en el área financiera, dando soporte en análisis de negocios, recursos humanos, compras, logística y servicios tecnológicos.

1.2.1.1. Misión

Ofrecer los mejores productos y servicios con una alta calidad que mejoren la calidad de vida de los consumidores actuales y futuros.

1.2.1.2. Visión

Ser reconocida como la mejor compañía de productos de bienes de consumo, generando satisfacción y valor mediante la calidad de sus productos.

1.2.1.3. Valores

Se ofrece una cultura única, buscando el crecimiento de los empleados tanto dentro del negocio como a nivel personal, uniendo la comunidad para la mejora continua de la vida de los consumidores. Los valores que se reflejan son comportamiento de una cultura de trabajo en equipo y un enfoque en la generación de valor. A continuación, se nombran los valores fundamentales para la empresa:

- Integridad
- Liderazgo
- Pasión por éxito
- Confianza

1.2.1.4. Equipo de trabajo

El equipo de trabajo en el que se desarrolla este proyecto es conocido como Ingeniería de Inteligencia Artificial, el cual es una organización que opera en Costa Rica. Esta se encuentra dentro de uno de los pilares de Tecnología de Información de la empresa, el cual es conocido como Data and Analytics o Datos y Analítica.

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

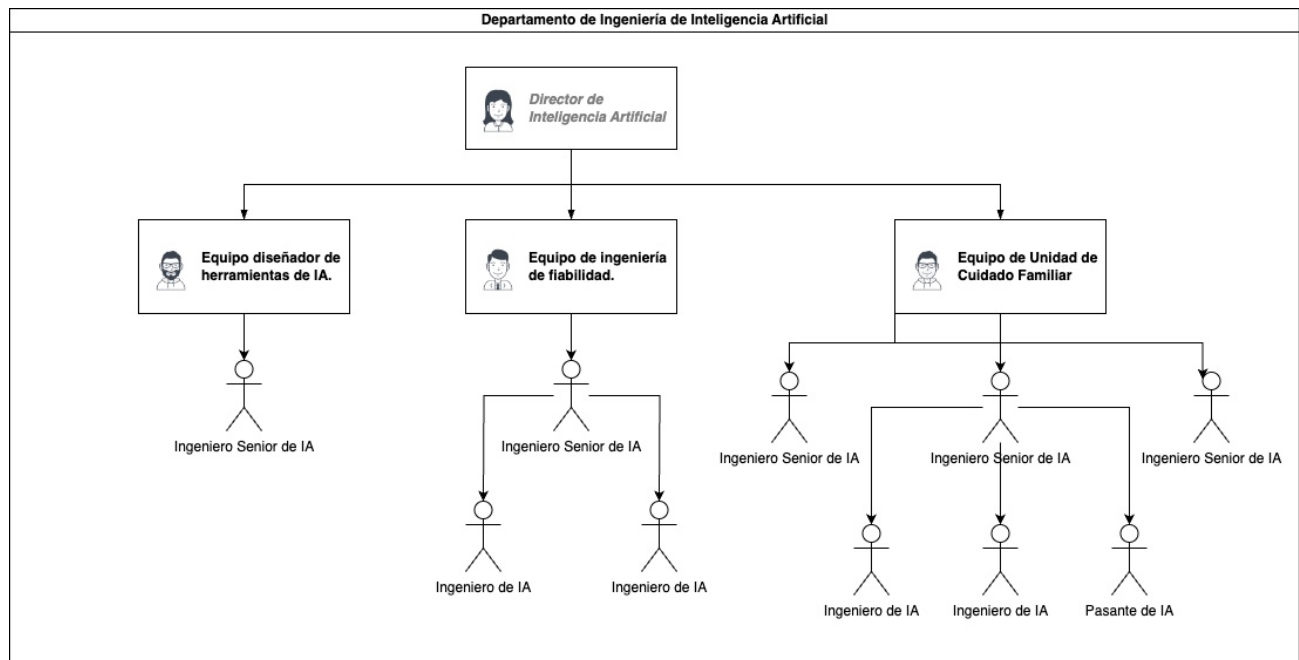
El equipo de Ingeniería de Inteligencia Artificial en Costa Rica se divide en tres secciones principales. La primera corresponde al equipo encargado del diseño de herramientas de IA, especialmente de AI Factory, desarrollando soluciones para la gestión y ejecución de proyectos relacionados con inteligencia artificial.

La segunda sección corresponde al equipo de Site Reliability Engineering (SRE). Su función es garantizar la estabilidad y confiabilidad de los proyectos en producción, mediante el uso de herramientas de automatización, monitoreo y gestión de *software*. Finalmente, la tercera sección brinda soporte a la Unidad de Cuidado Familiar, la cual se especializa en ofrecer soluciones tecnológicas mediante ingeniería de datos, análisis de datos, ciencia de datos y optimización de procedimientos de procesos de manufactura.

Como desarrollador de este proyecto y pasante en el equipo de Ingeniería de Inteligencia Artificial, el autor de este trabajo forma parte de la sección que brinda servicios a la Unidad de Cuidado Familiar, donde se encarga de operacionalizar proyectos, ingeniería de datos y automatización de procedimientos.

A continuación, se presenta el organigrama de la organización de Ingeniería de Inteligencia Artificial de la empresa manufacturera.

Figura 1: Organigrama de la organización.



Fuente: elaboración propia.

1.2.2. Trabajos similares realizados dentro y fuera de la organización

1.2.2.1. Proyectos internos de la organización

A continuación, se describen tres proyectos realizados dentro de la organización, los cuales sirven de insumo para el desarrollo de este proyecto:

- En la organización de Ingeniería de Inteligencia Artificial se inició un proyecto en 2021 para la estandarización y experimentación de modelos de ciencias de datos, conocido como *AI Factory*. Este implementó una metodología unificada para el desarrollo eficiente de LLM, permitiendo su implementación y supervisión a gran escala. Además, facilitó la creación y despliegue de modelos, optimizando el rendimiento de las soluciones a nivel de ciencia de datos. Cabe destacar que fue desarrollado para AI Factory y busca estandarizar diferentes prácticas, tecnologías y procesos para el desarrollo de proyectos que incorporan modelos de ciencias de datos. Asimismo, tiene como objetivo la integración con datos en la nube, abarcando una amplia variedad de tecnologías para la implementación de CI/CD en sus soluciones. Actualmente, este proyecto sigue en desarrollo, adoptando más funcionalidades para la integración de estas soluciones computacionales.
- El proyecto *Streamlined Operation Techniques for AI Engineers* busca establecer prácticas operativas eficientes para el desarrollo y despliegue de soluciones de IA de alta calidad. Su objetivo es mejorar la colaboración, documentación y el uso de diversas herramientas y plataformas. Además, su enfoque se centra en soluciones aplicables con CI/CD, entornos de desarrollo y producción, Databricks Asset Bundles, Azure ML, entre otros. Este sigue desarrollándose en el presente, conformando una de las referencias destacadas para el Trabajo Final de Graduación.
- El proyecto *Planificación de Capacidad Rough Cut* es una herramienta de planificación de capacidad diseñada para equilibrar la capacidad mensual de productos en la Unidad de Cuidado Familiar. Actualmente, este proyecto de manufactura se encuentra en proceso de operacionalización y automatización de procesos, convirtiéndose en un insumo clave para la prueba de concepto de este trabajo, aplicando la metodología DevOps CI/CD. En el presente, este proyecto sigue recibiendo requerimientos de ingeniería de datos, en proceso a operacionalizar las funcionalidades de este.

1.2.2.2. *Proyectos externos de la organización*

A continuación, se describen los proyectos externos a la organización que sirven como insumo para el desarrollo de este trabajo:

- **Trabajo Final de Graduación de Cristófer Gómez Márquez (2024):** se toma como referencia el Trabajo Final de Graduación realizado en el Instituto Tecnológico de Costa Rica en la Escuela de Administración de Tecnología de Información. Este consistió en una propuesta de metodología para la gestión de desarrollo de *software* en una empresa de tecnología, basada en las mejores prácticas de la industria. Su enfoque se fundamentó en la investigación-acción, metodología similar a la que se busca adoptar en este trabajo de graduación. Este estudio fue desarrollado para evaluar la gestión de proyectos de la organización en la cual se realizaba, elaborando una metodología que se adecuara a la realidad de la empresa en cuestión.
- **Trabajo Final de Graduación de Kenneth Martínez Avendaño (2018):** este proyecto titulado: *Metodología de Administración de Proyectos para una empresa de Desarrollo de Software* fue desarrollado en la Escuela de Administración de Tecnología de Información del Instituto Tecnológico de Costa Rica, para optar por el grado de Licenciatura. Este documento describió una investigación de tipo aplicada, que buscó describir y analizar el proceso de la gestión de proyectos de la organización. Además, desarrolló de formas aplicadas, las comparaciones del proceso de gestión de proyectos de la empresa, enfocándose en la iniciación, planificación, ejecución, monitoreo y control de los procedimientos, tomando como marco de referencia el *framework* Scrum, para su enfoque en las metodologías ágiles.
- **Trabajo Final de Graduación de Verónica Vargas Mora (2018):** este proyecto fue realizado en la Escuela de Administración de Tecnología de Información para optar por el grado de Licenciatura. Se enfocó en la gestión de proyectos de soluciones tecnológicas, proponiendo un enfoque para el desarrollo e implementación de procesos que anteriormente se realizaban de manera inconsistente, generando problemas de control y seguimiento dentro de la organización. Además, se identificaron las necesidades y limitaciones pertinentes de la empresa de soluciones integrales S.A. y, posteriormente, se elaboró una propuesta de metodología estándar para la gestión de proyectos que se adecuara a la realidad de la empresa.

1.3. Planteamiento del problema

En esta sección, se describe la situación problemática hallada dentro del entorno de la organización, la cual motiva el desarrollo del proyecto, así como se hace la mención de los beneficios esperados del producto.

1.3.1. Situación problemática

Actualmente, el equipo de Ingeniería de Inteligencia Artificial, específicamente el encargado de brindar servicios a la Unidad de Cuidado Familiar, opera en Costa Rica desde hace tres años, brindando servicios en la gestión de proyectos de tecnologías de información. Sin embargo, no se han integrado correctamente prácticas de mejora continua, flujos de trabajo ni políticas para el desarrollo de proyectos de esta índole, considerando tanto estándares de calidad internos como externos. En particular, no se aplica el AI Readiness Criteria, un conjunto de políticas, estándares y criterios diseñados para garantizar que los procesos y soluciones de TI sean robustos, seguros y operativamente correctos.

La falta de alineación con el AI Readiness Criteria ha generado deficiencias en la confiabilidad de los procesos, el control de cambios, la documentación y la integración con el AI Factory de la empresa; lo que impacta en la entrega e integración del *software* (CI/CD). Como resultado, el equipo de Ingeniería de Inteligencia Artificial carece de un proceso estandarizado para la gestión de proyectos y metodologías de mejora continua, afectando la eficiencia operativa y, por ende, el valor y calidad de los proyectos.

En el presente, el equipo de Ingeniería de Inteligencia Artificial no cuenta con un proceso estandarizado para la gestión de proyectos E2E en la unidad de negocio Cuidado Familiar. Existe una ausencia de un proceso definido para la aplicación de DevOps CI/CD dentro del equipo, a pesar de su importancia en la industria de tecnologías de información para garantizar el cumplimiento de las mejores prácticas en el desarrollo de *software* y mantener la competitividad organizacional.

Además, la deficiencia de estandarización en los componentes de desarrollo, la obtención de datos, dificultades de integración y el acceso a la información, así como la ausencia de un entorno de producción han obstaculizado la adopción de los criterios, políticas y mejores prácticas tanto internas como externas de la empresa.

Actualmente, los proyectos se ejecutan sin una visión unificada ni la automatización esperada de un equipo de ingeniería de inteligencia artificial. Esto afecta directamente el cumplimiento de métricas, eficiencia, tiempo de trabajo y la competitividad del equipo.

Sumado a lo anterior, el equipo de Ingeniería de Inteligencia Artificial no cuenta con un proceso definido para el desarrollo de proyectos en conjunto con el facilitador de Tecnología y Datos dentro de la plataforma Databricks. Para gestionarlos, el equipo utiliza un Launchpad, una plataforma integral diseñada para el desarrollo de proyectos en Databricks. Actualmente, la mayor parte de la información de las plantas manufactureras se consume a través de este Launchpad. Por esta razón, el equipo debe utilizar esta plataforma para la exploración y experimentación de los datos de la unidad de Cuidado Familiar.

Este servicio proporciona herramientas preconfiguradas para el desarrollo de *pipelines* y soluciones computacionales en Databricks, con un enfoque en los datos del proceso de manufactura. Sin embargo, el principal problema radica en la falta de un puente de comunicación entre el facilitador de datos (los encargados del Launchpad) y el equipo de Ingeniería de Inteligencia Artificial.

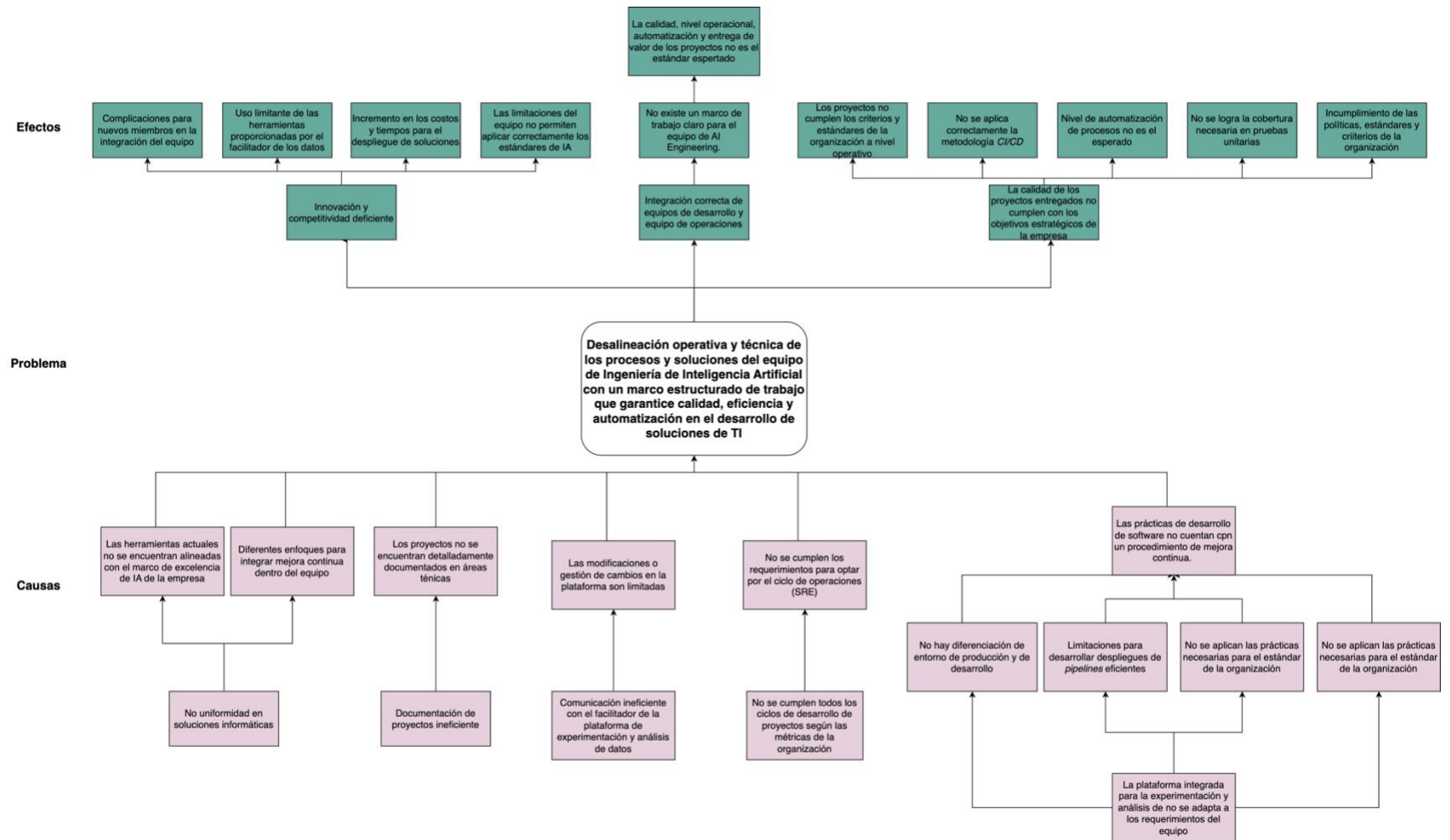
Es importante destacar que el Launchpad no cubre algunas necesidades clave del equipo de Ingeniería de Inteligencia Artificial, como la integración de componentes específicos de IA, específicamente con el AI Factory de la compañía, el cumplimiento con los estándares de seguridad (Infosec), la flexibilidad para la instalación de dependencias técnicas y la compatibilidad con herramientas de Databricks. De igual forma, existen limitaciones en la implementación de CI/CD, la ausencia de un entorno de producción y la falta de conexión con el equipo de operaciones (SRE), lo que dificulta la escalabilidad de los proyectos.

Estas limitaciones generan problemas en la gestión de datos y dificultan la aplicación de los criterios de aceptación establecidos por la organización de Ingeniería de Inteligencia Artificial, afectando la eficiencia y la alineación de los proyectos con los estándares de la empresa.

Hoy en día, los proyectos no se desarrollan con un marco de trabajo estandarizado debido a las limitantes de la plataforma, así como problemas persistentes de documentación y planificación. Como consecuencia, el equipo de Inteligencia Artificial no logra implementar de manera efectiva los procesos del marco de CI/CD, ya que las herramientas y recursos disponibles no facilitan la integración de estas metodologías en el desarrollo de los proyectos del equipo. Esto conlleva a un desarrollo inconsistente de soluciones y enfoques que no cumplen con los criterios y estándares de la organización de Ingeniería de Inteligencia Artificial.

1.3.1.1. *Árbol del problema*

Figura 2: *Árbol del problema*



Fuente: elaboración propia.

1.3.2. Justificación del proyecto

En el contexto actual de transformación digital, la gestión eficiente de proyectos de Ingeniería de Inteligencia Artificial es un pilar importante para las empresas manufactureras que buscan mantener su competitividad en términos de TI. Sin embargo, el equipo de Ingeniería de Inteligencia Artificial enfrenta desafíos críticos que limitan su capacidad para la entrega de soluciones robustas y alineadas con los estándares de la empresa. La ausencia de un marco de trabajo estandarizado ha generado una brecha entre las prácticas actuales y los requisitos del AI Readiness Criteria, lo que afecta la calidad, escalabilidad y sostenibilidad de las soluciones implementadas.

Ante este escenario, la propuesta de un marco de trabajo estandarizado basado en DevOps CI/CD se basa en una solución integral. Como señalan González y Calderón (2023): “Las metodologías ágiles y *CI/CD* permiten a las empresas generar mayor valor en sus desarrollos”; algo que el equipo no logra conseguir debido a la falta de comunicación entre el facilitador del Launchpad, sus procesos técnicos y su uniformidad de soluciones informáticas.

Esta propuesta marco de trabajo no solo abarcaría las deficiencias actuales, sino que también permite:

1. **Automatizar etapas críticas en el desarrollo:** ejecución de pruebas y despliegue de soluciones efectivas, reduciendo tiempos de entrega y minimizando errores humanos.
2. **Alinear los proyectos con el AI Readiness Criteria:** asegurando el cumplimiento de políticas y estándares de calidad, documentación clara, control de cambios, desarrollo eficiente, pruebas unitarias y correcta integración con *CI/CD*.
3. **Fortalecer la conexión con el equipo de SRE y el AI Factory:** facilitando entornos de producción que garanticen la estabilidad y escalabilidad de proyectos mediante tecnologías relacionadas a la Ingeniería de Inteligencia Artificial.

Como menciona Mendoza Torres (2023): “*DevOps CI/CD* integra todos los involucrados de una implementación de software con el objetivo de desarrollar software de alta calidad, con integridad y seguridad”. Esta integración es vital para resolver la desconexión actual entre el equipo de Ingeniería de Inteligencia Artificial, el facilitador de Launchpad en Databricks, los criterios de completitud de calidad de *software* y los estándares de seguridad.

Finalmente, el marco propuesto se alinea con las áreas temáticas clave de la Escuela de Administración de Tecnologías de Información del Tecnológico de Costa Rica, abordando:

- **Administración de Proyectos de TI:** implementación de las mejores prácticas ágiles de la industria, con un marco de trabajo basado en DevOps *CI/CD* para la gestión de proyectos de Ingeniería de Inteligencia Artificial.

- **Gestión de tecnología de información:** estandarización de procesos y herramientas tecnológicas alineadas con las necesidades actuales del equipo.
- **Gestión de datos e información:** optimización de la interacción del equipo con el facilitador del Launchpad, mediante una estrategia clara que mejore la gestión y aprovechamiento de los datos en el desarrollo de proyectos.
- **Análisis, diseño y programación de sistemas de información:** aplicación de metodologías ágiles y automatización para la mejora de la calidad y escalabilidad de soluciones informáticas.

Este trabajo no solo propone una solución a una necesidad operativa, sino que también demuestra cómo la estandarización de procesos, respaldada de la metodología DevOps, puede transformar la gestión de proyectos de Ingeniería de Inteligencia Artificial, enfocado en aquellos de manufactura. Al reducir la brecha entre herramientas, estándares y equipos, se puede incrementar la competitividad del equipo, asegurando soluciones más escalables y alineadas con los desafíos tecnológicos actuales.

1.3.3. Beneficios esperados o aportes del Trabajo Final de Graduación

1.3.3.1. Beneficios directos

- **Implementación de CI/CD para optimizar el desarrollo de proyectos:** la propuesta de la metodología CI/CD dentro del marco de trabajo DevOps optimiza el desarrollo de proyectos de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar, integrando iteraciones frecuentes y controladas en la integración y validación del código. Esto permite establecer un entorno de trabajo ágil, donde los cambios sean implementados y validados de manera continua, asegurando entregas de mayor calidad y alineadas con los requerimientos de negocio. Esto abarca procedimientos de automatización, análisis de datos, gestión y visualización de estos, en el área de manufactura de esta unidad de negocio, visualizando la automatización de procedimientos y eficacia de estos.
- **Fortalecimiento de la colaboración con el facilitador del Launchpad de Databricks:** la propuesta del marco de trabajo basado en DevOps CI/CD permite alinear las necesidades del equipo de Ingeniería de Inteligencia Artificial con las capacidades y requerimientos del facilitador del Launchpad de Databricks. A través de la definición de estándares operativos, incorporación de herramientas clave y una mejor integración de datos, se establece un proceso claro que facilite la colaboración y comunicación entre ambas partes, con el objetivo de resolver las limitaciones técnicas y operativas existentes.

- **Identificación de limitaciones y oportunidades de mejora del flujo de trabajo:** la propuesta del marco de trabajo DevOps CI/CD permite un análisis del funcionamiento actual del equipo de Ingeniería de Inteligencia Artificial. Esto facilita la identificación de sus principales limitaciones. Además, este enfoque busca reconocer acciones correctivas para la optimización de procesos y garantizar soluciones más eficientes, escalables y alineadas con los estándares organizacionales.

A continuación, se presentan los indicadores clave que permiten evaluar el impacto de la propuesta del marco de trabajo basado en DevOps CI/CD en el equipo de Ingeniería de Inteligencia Artificial:

- Reducción en los tiempos de entrega de despliegues, optimizando el ciclo de vida de los proyectos.
- Incremento en la frecuencia de despliegues.
- Reducción del porcentaje de errores críticos.
- Integración de *pipelines* de CI/CD dentro de la prueba de concepto.
- Análisis y beneficios de una propuesta de DevOps CI/CD, con el enfoque de mejorar la comunicación con el facilitador del Launchpad de Databricks.

1.3.3.2. *Beneficios indirectos*

- Mejora continua en el desarrollo y despliegue de soluciones informáticas: el marco de trabajo basado en CI/CD proporciona a los desarrolladores un entorno de mejora continua (Continuous Improvement – CI), minimizando el riesgo de errores en el código y optimizando procesos de desarrollo y despliegue de soluciones. Este enfoque reduce la necesidad de retrabajo, mejora la calidad del software entregado y permite al equipo identificar errores de manera temprana.
- Al implementar una política de estandarización de procesos, se presenta una gestión de cambios dentro del desarrollo de los proyectos. El equipo implementaría mejores prácticas que garanticen proyectos iterativos y con mayor estabilidad en su ciclo de vida.
- La mejora del rendimiento del equipo es uno de los enfoques más importantes que las organizaciones hoy en día deben desarrollar. Con la propuesta del marco de trabajo, se busca incrementar la competitividad a nivel organizacional, un proceso de integración continuo y fluido que permita generar soluciones más efectivas a nivel de ingeniería de datos, modelos de ciencia de datos, así como de soluciones en el ámbito de Inteligencia Artificial Generativa.

- Aumento de satisfacción con el negocio, brindando proyectos con un mayor índice de disponibilidad, escalamiento y mejores posibilidades de mitigación ante problemas. Esto incrementa la percepción positiva que se tiene sobre el equipo de Inteligencia Artificial.
- Disminución de solicitudes de soporte técnico relacionado al Launchpad, por lo tanto, reduciendo tiempos de espera para el desarrollo de soluciones.

1.4. Objetivos del Trabajo Final de Graduación

1.4.1. Objetivo general

Desarrollar una propuesta de marco de trabajo que estandarice la gestión de proyectos del equipo de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar, mediante la metodología DevOps CI/CD, con el propósito de mejorar los procesos de planificación, documentación, análisis, diseño, pruebas y despliegue de proyectos, asegurando su alineación con las mejores prácticas de la industria, durante el primer semestre del 2025.

1.4.2. Objetivos específicos

1. Analizar la situación actual del equipo de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar, con el fin de identificar los requerimientos técnicos y operativos necesarios para implementar correctamente la metodología DevOps CI/CD.
2. Diseñar el marco de trabajo basado en la metodología DevOps CI/CD, a partir de los requerimientos técnicos y operativos identificados en el análisis de la situación actual, asegurando la adopción de las mejores prácticas ágiles que sean adaptables al desarrollo de proyectos del equipo de Inteligencia Artificial.
3. Comparar las diferencias operacionales en términos de eficiencia, tiempos de entrega y calidad de las soluciones informáticas de la gestión de proyectos de Inteligencia Artificial entre la situación actual y la propuesta de estandarización basada en DevOps CI/CD, con el propósito de validar las mejoras implementadas en el ciclo de vida de los proyectos.
4. Elaborar una prueba de concepto en un proyecto representativo de la Unidad de Cuidado Familiar, priorizando la integración continua, automatización de pruebas, despliegue de soluciones e integración de CI/CD, evaluando el ajuste de la propuesta de estandarización del marco de trabajo de gestión de proyectos.

1.5. Alcance

El alcance definido de esta propuesta radica en el desarrollo de un marco de trabajo estandarizado para el equipo de Inteligencia Artificial, el cual basa sus metodologías en la estructura de DevOps CI/CD. Este marco de trabajo busca abordar las necesidades de planificación, documentación, análisis, desarrollo, pruebas y despliegue de proyectos, alineándose con las mejores prácticas de la industria.

A continuación, se detallan las fases y elementos que se encuentran dentro y fuera del alcance del proyecto:

Tabla 1: Fases del proyecto según el alcance

Fases del proyecto	Descripción
Análisis de la situación actual	<ul style="list-style-type: none">• Identificación de las necesidades técnicas y operativas para el ciclo de vida de los proyectos del equipo de Inteligencia Artificial. Visualizando las diferentes implementaciones que se realizan en el equipo actualmente.• Identificación de las limitaciones en las funcionalidades implementadas actualmente en la plataforma Databricks.• Análisis de las herramientas que permitan la ejecución correcta de la metodología de DevOps CI/CD, las cuales se adapten correctamente a las necesidades del equipo, específicamente, la integración con Databricks y su marco de trabajo, con tecnologías tales como GitHub, SonarQube y AI Factory.
Diseño del marco de trabajo	<ul style="list-style-type: none">• Definición de los procesos operativos y técnicos prioritarios para el desarrollo de un <i>framework</i> funcional basado en la metodología CI/CD.• Incorporación de la metodología DevOps CI/CD a nivel E2E de los proyectos del equipo de Inteligencia Artificial a nivel de planificación, desarrollo y construcción de solución, testeado de la solución y despliegue de soluciones.• Establecimiento de las mejores prácticas a nivel de metodologías ágiles que sean aplicables para el desarrollo de proyectos de calidad del equipo de Inteligencia Artificial.• El diseño de este busca abarcar los procesos necesarios para cada fase del proyecto, incluyendo documentación, planificación, gestión y despliegue operativo del mismo, así como el desarrollo de herramientas recomendadas para la implementación de metodologías ágiles mediante DevOps y CI/CD.

Fases del proyecto	Descripción
Diseño del marco de trabajo	<ul style="list-style-type: none">• Definición de los procesos operativos y técnicos prioritarios para el desarrollo de un <i>framework</i> funcional basado en la metodología CI/CD.• Incorporación de la metodología DevOps CI/CD a nivel E2E de los proyectos del equipo de Inteligencia Artificial a nivel de planificación, desarrollo y construcción de solución, testeo de la solución y despliegue de soluciones.• Establecimiento de las mejores prácticas a nivel de metodologías ágiles que sean aplicables para el desarrollo de proyectos de calidad del equipo de Inteligencia Artificial.• El diseño de este busca abarcar los procesos necesarios para cada fase del proyecto, incluyendo documentación, planificación, gestión y despliegue operativo del mismo, así como el desarrollo de herramientas recomendadas para la implementación de metodologías ágiles mediante DevOps y CI/CD.
Comparación entre la situación actual y la propuesta de estandarización	<ul style="list-style-type: none">• Comparación del marco de trabajo actual con la propuesta desarrollada, evaluando los cambios realizados a nivel de tecnologías y metodología.• Comparación de la propuesta actual mediante métricas de desempeño, tiempos de despliegue, frecuencia de errores y estabilidad de las soluciones.
Elaboración de prueba de concepto	<ul style="list-style-type: none">• Operacionalización de la prueba de concepto de un proyecto representativo de la Unidad de Cuidado Familiar.• Generación de informe sobre el uso de marco de trabajo en un proyecto representativo de la Unidad de Cuidado Familiar, implicando el ciclo de vida según la metodología DevOps CI/CD.• Representación mediante diagramas, visualización de las actividades del marco de trabajo, así como resultados obtenidos con la prueba de concepto.

Fuente: elaboración propia.

1.6. Supuestos

En la presente sección, se describen los factores y elementos que se asume se cumplirán en el desarrollo del proyecto.

- Se cuenta con el acceso a herramientas de tecnología y plataformas necesarias, tales como Databricks, GitHub, SonarQube y Power BI, asegurando su disponibilidad para el uso del proyecto.
- Se asume la comunicación efectiva entre los facilitadores de la gestión de datos en Databricks, para la identificación de limitaciones y necesidades del equipo de Inteligencia Artificial.
- Disponibilidad del equipo de trabajo de Ingeniería de Inteligencia Artificial, disponibles para participar en actividades del proyecto, incluyendo reuniones, entrevistas y talleres de recolección de datos.
- La estimación del alcance es la adecuada para la realización del proyecto.

1.7. Entregables

A continuación, se describen los entregables del proyecto, basándose específicamente en las fases y en los objetivos de este.

- Análisis de la situación actual: el entregable correspondiente al primer objetivo es el diagnóstico inicial de la problemática en la falta de regulación de los proyectos del equipo de Ingeniería de Inteligencia Artificial. Este diagnóstico es de carácter documental, basándose en las necesidades técnicas y operativas identificadas en el proceso actual del desarrollo de proyectos. De igual forma, incluye un análisis del flujo de trabajo actual, así como en las limitaciones de la plataforma de gestión de datos brindada en Databricks. Adicionalmente, se brinda un diagrama causa-efecto, que represente las diferentes limitantes para la regulación de soluciones implementadas para los proyectos desarrollados por el equipo.
- Diseño del marco de trabajo: se realiza una guía del marco de trabajo mediante la metodología DevOps CI/CD aplicable a las necesidades identificadas previamente. Se detalla un documento estructural, el cual defina los procesos operativos y técnicos que se deben implementar en el marco de trabajo, incluyendo los procesos de planificación, desarrollo, testeo y despliegue. De igual forma, se incluye el marco de referencia con las herramientas y prácticas para la implementación del marco de trabajo.

- Comparativa entre la situación actual y la propuesta desarrollada: se elabora un informe, el cual detalla las diferencias entre la situación actual y la solución propuesta. Este informe está respaldado por métricas que reflejan la eficiencia en aspectos como tiempos de despliegue, metodologías ágiles, frecuencia de errores y estabilidad de las soluciones propuestas.
- Elaboración de la prueba de concepto: documento que ejemplifica los resultados obtenidos al implementar el marco en un proyecto de la Unidad de Cuidado Familiar, donde se destacan los beneficios implementados, herramientas y prácticas implementadas para el desarrollo del marco de trabajo realizado a partir de la implementación de DevOps y CI/CD. De igual forma, se plantean diagramas del flujo de actividades de la prueba de concepto con resultados y métricas obtenidas.

1.8. Limitaciones

En esta sección, se limitan los factores principales que limitan el alcance del proyecto, siendo una restricción ante procedimientos técnicos u organizacionales.

- La propuesta del marco de trabajo se enfoca únicamente en una unidad de negocio de la organización de Ingeniería de Inteligencia Artificial, la cual se conoce como Cuidado Familiar. Por esta razón, no se debe evaluar la problemática como una organización, sino únicamente como un problema dentro de una unidad representativa de esta.
- La etapa de monitorización visualizada en la metodología DevOps no será abarcada en el alcance de este trabajo, esto quiere decir que se evalúa la propuesta desde la planificación hasta el despliegue de proyectos.
- El proyecto no contempla el desarrollo de nuevas funcionalidades en Databricks ni la adquisición de nueva tecnología o plataformas para el desarrollo de este. Esto quiere decir que se debe asumir el entorno de trabajo actual para la comparación con el marco de trabajo que se desea proponer.
- Las políticas de privacidad de la organización indican que no será posible administrar archivos, códigos o datos reales de la organización, ni ningún tipo de información que implique una relación con dicha empresa.
- Existe una escasa documentación del proceso actual realizado por el equipo de Ingeniería de Inteligencia Artificial.
- La disponibilidad del equipo no es de carácter inmediato.

2. Marco Conceptual

El marco conceptual tiene como finalidad proporcionar un sistema coordinado de los conceptos, definiciones, teorías y metodologías que se desarrollan en la investigación. Este marco permite contextualizar el problema tanto en el ámbito práctico como en el teórico, integrando conocimientos previos y organizándolos para la construcción de la propuesta de solución.

En la **Figura 3**, se muestra la estructura del marco conceptual, este permite la visualización de los conceptos según su ámbito y utilidad en el proyecto.

Figura 3: Estructura del Marco Conceptual



Fuente: elaboración propia.

2.1. Conceptos prácticos y teóricos

2.1.1. Gestión de la calidad

La gestión de calidad en los productos se basa en la constancia, asegurando eficiencia, confiabilidad y mejora continua en entornos tecnológicos. Este concepto se crea en la idea de crear valor de forma sostenible y continua para el cliente, incorporando prácticas que permitan corregir errores de forma temprana y prevenir su ocurrencia.

Este concepto se relaciona con integraciones o marcos establecidos como Lean y DevOps; asimismo, con el modelo de PDCA (Plan-Do-Check-Act). Este ciclo permite establecer un proceso iterativo para la identificación de mejoras y la aplicación de correcciones en simultáneo. Este enfoque permite a los equipos mejorar continuamente sus procesos de retroalimentación y el aprendizaje iterativo, permitiendo el fortalecimiento de la calidad organizacional (Kim et al., 2021).

La metodología DevOps, al conceptualizar sus bases en el enfoque Lean y su flujo de valor, refuerza los principios de reducir desperdicios, optimizar flujos de trabajo y garantizar que cada actividad se base en el valor agregado. Dentro del enfoque Lean, se define el concepto de flujo de valor, el cual, según *The DevOps Handbook*, es: “Una secuencia de actividades necesarias para diseñar, desarrollar y entregar un producto o servicio al cliente, incluyendo el flujo de materiales como el de información. Identificar, mapear y optimizar estos flujos permite reducir tiempos de espera, eliminar desperdicios y acelerar la entrega de valor” (Kim et al., 2021).

Es destacable como la metodología DevOps puede integrar la gestión de calidad fomentando la colaboración entre áreas prácticas enfocadas en la automatización de tareas repetitivas, detectar cuellos de botella y, por ende, optimizando el flujo de valor, el cual corresponde a una herramienta clave en la gestión de la calidad de proyectos de *software*.

2.1.2. Metodologías ágiles

Las metodologías ágiles han transformado la gestión de proyectos de tecnología, donde el enfoque se basa en la colaboración entre equipos, procedimientos eficientes, entregas tempranas de valor y la capacidad de adaptación a nuevos procedimientos de negocio. Primeramente, se debe remontar a los principios de esta metodología, el cual es conocido como *The Agile Manifesto* o el Manifiesto ágil, el cual, según *The DevOps Handbook* es: “El objetivo del Manifiesto ágil fue crear un conjunto de valores y principios que pudieran capturar las ventajas de métodos de desarrollo más adaptativos, en comparación con los procesos tradicionales predominantes en ese momento, tales como el modelo cascada y metodologías como el *Rational Unified Process (RUP)*. El término *ágil* es reconocido por aumentar de forma notable productividad y la capacidad de respuesta múltiples organizaciones de desarrollo de software” (Kim et al., 2021).

Según el PMBOK Guide: “Las metodologías ágiles promueven la entrega incremental, el Feedback constante y la capacidad de adaptación rápida al cambio, promoviendo la transparencia y mejora de la capacidad de aprendizaje del equipo a lo largo del proyecto” (Project Management Institute, 2021). En este contexto, Scrum ha sido una de las metodologías más adoptadas, facilitando el trabajo en equipo mediante ciclos de duración denominados *sprints*, fomentando la entrega eficiente y la mejora continua.

En referencia a este, en la estandarización de un marco de trabajo basado en DevOps, las metodologías ágiles representan una oportunidad estratégica para estructurar y dinamizar proyectos de inteligencia artificial basada en prácticas relacionadas directamente con el concepto de “ágil”.

2.1.3. DevOps

DevOps es una metodología, la cual busca promover la colaboración entre equipos que normalmente se encuentran aislados, ya sea en desarrollo de *software* (*Dev*), en las operaciones (*Ops*), aseguramiento de la calidad (*QA*) y seguridad de la información (Infosec). Este enfoque basado en metodologías ágiles busca desarrollar mayor velocidad, calidad y estabilidad en las entregas de *software* proporcionadas por el equipo.

Según Kim et al. (2021) de *The DevOps Handbook*, se pueden identificar tres principios fundamentales para el desarrollo de DevOps, y que este funcione para crear agilidad, rentabilidad y seguridad en organizaciones de tecnología:

1. Incrementación del flujo de trabajo desde el desarrollo hasta producción.
2. Ampliar la retroalimentación y el *feedback* constante para la detección de problemas y hacer correcciones rápidamente.
3. Promover una cultura de aprendizaje y mejora continua.

El enfoque DevOps permite, mediante estas pautas, desarrollar, probar y desplegar código de forma independiente, segura y confiable, construyendo un sistema de trabajo seguro, el cual se enfoca en la productividad del equipo y, por lo tanto, en la satisfacción del cliente.

DevOps, además de tener un enfoque en proyectos de tecnología, basa sus principios en metodologías Lean. Según Forsgren et al. (2018), se destaca que las prácticas DevOps se encuentran directamente relacionadas con un mejor desempeño técnico y de negocio. Particularmente, se compone de cuatro métricas clave para la evaluación efectiva de equipos integrados en entornos DevOps:

1. **Lead Time:** tiempo en el cual el código se encuentra en *commit* hasta su integración en producción.
2. **Deployment Frequency:** frecuencia de despliegues exitosos.
3. **Time to Restore Service:** tiempo promedio de recuperación ante alguna falla.
4. **Change Fail Rate:** proporción de cambios en producción que generan fallos.

Estas métricas permiten cuantificar el impacto real de la metodología DevOps sobre aspectos del equipo, ya sea de eficiencia, estabilidad o innovación organizacional. Asimismo, permite la identificación de cuellos de botella y oportunidades de mejora con mayor facilidad (Forsgren et al., 2018).

2.1.4. Continuous Integration, Continuous Delivery y Continuous Deployment

El enfoque de Integración Continua (CI) y Entrega Continua (CD) corresponde a uno de los pilares más importantes dentro de la metodología DevOps. Estas prácticas en conjunto permiten acelerar el flujo de trabajo desde el desarrollo de código hasta su fase de entrega en producción, con el fin de reducir errores humanos, aumentar la calidad del *software* y mejorar la frecuencia de las entregas realizadas.

Primeramente, se encuentra el *Continuous Integration*, el cual, según *The DevOps Handbook*: “Se basa en la práctica de integrar cambios de código frecuentemente y verificar mediante pruebas automatizadas que el código no incurra en errores” (Kim et al., 2021).

Por otro lado, se encuentra la *Continuous Deployment*, el cual, dentro del libro de *The DevOps Handbook*: “Se basa en asegurar que el software se encuentre en estado desplegable en todo momento, y que cualquier cambio que sea aprobado pueda ser ejecutado y desplegado en producción de maneja segura y automatizada” (Kim et al., 2021).

Continuous Delivery implica que el *software* siempre se mantenga en un estado estable y listo para su integración en producción. Según el libro *Continuous Integration and Continuous Deployment*: “*Continuous Delivery* se basa en el hecho de que siempre debe haber una rama principal estable del Código, y el despliegue a producción puede realizarse desde cualquier rama” (Van Merode, 2023). Esto asegura que los cambios aprobados pueden desplegarse según corresponda, con procesos consistentes de validación que garantizan la ejecución tanto en ambientes de prueba como en producción.

El concepto de *Continuous Deployment* busca la configuración de diferentes capacidades que permiten la realización de cambios de cualquier tipo. Esto corresponde para la configuración de cambios, arreglo de *bugs* y experimentos dentro del estado de producción, con el fin de un despliegue sostenible y rápido.

Para la implementación correcta del concepto de *Continuous Deployment*, se deben seguir los siguientes pilares:

1. **Gestión integral de la configuración:** la posibilidad de crear ambientes de desarrollo para la configuración de la infraestructura, control de versiones, dependencias, *testing* y despliegue de soluciones.
2. **Continuous Integration:** además de ser una práctica para aumentar la frecuencia de las integraciones en el código en el día, busca la integración con el concepto de trabajar en *small batches* e implementar pruebas unitarias, gestión de procesos fallidos y retroalimentación constante.

3. **Continuous Testing:** se refiere al contexto del *testing*, sin embargo, su aplicación es diferente no únicamente al final del desarrollo, sino que de manera constante durante todo el ciclo de vida de este.

Para el funcionamiento correcto de CI/CD dentro de un proyecto, son necesarias distintas herramientas que se encuentran encapsuladas dentro de un *pipeline*. La construcción de un *pipeline*, según el libro *Accelerate* es: “Un proceso automatizado que se basa en las fases de construcción, pruebas, integración, despliegue y validación del software, destacando prácticas que mejoren el rendimiento técnico, estabilidad del servicio y la innovación organizacional” (Forsgren et al., 2018).

A continuación, se detallan los elementos técnicos fundamentales para la implementación de CI/CD:

1. Uso de control de versiones para el código.
2. Aplicación de pruebas automatizadas en cada etapa del *pipeline*.
3. Adopción de un enfoque en *small batches*, lo cual permite ciclos de retroalimentación y verificación de código de una forma más rápida y efectiva.
4. Prácticas como el *trunk-based development*, donde se maximiza y favorece la integración continua mediante el control del número de ramas activas.
5. Movilización de actividades críticas de seguridad y validación del proyecto a etapas tempranas del ciclo de vida del *software*.

Implementar CI/CD dentro de los proyectos de *software* y como parte de la metodología de DevOps, permite medir el rendimiento de la entrega de *software* mediante sus cuatro métricas identificadas previamente, posibilitando a las organizaciones más competitividad y adaptabilidad.

2.1.5. Gestión de proyectos de IA

En pleno auge de la inteligencia artificial, se han conformado equipos especializados en el desarrollo, integración y escalabilidad de soluciones informáticas tanto en empresas tecnológicas como no tecnológicas. Además, la gestión de proyectos de inteligencia artificial presenta retos particulares debido a la experimentación y las múltiples disciplinas que se deben desarrollar para este tipo de soluciones.

En el marco de la transformación digital, esta no solo implica adoptar herramientas, sino alinear personas, procesos y tecnología bajo un modelo de entrega constante de valor (Freixas, 2022). En este contexto se conceptualiza el AI Readiness Criteria, siendo un elemento clave que asegura la madurez de un proyecto de Ingeniería de Inteligencia Artificial durante su desarrollo y su integración en producción. Los criterios más importantes desarrollados en este concepto son:

1. Documentación extensa, estandarizada y detallada a nivel técnico.
2. Control de versiones tanto de soluciones como de modelos integrados.
3. Estándares de seguridad y de cumplimiento.

La búsqueda de gestión efectiva de proyectos de inteligencia artificial, así como la implementación de prácticas modernas y eficientes de ingeniería, convierten el AI Readiness Criteria en una herramienta fundamental. Esta permite evaluar el grado de alineación del proyecto con los estándares organizacionales, promoviendo una visión integral que articula calidad técnica, seguimiento de los objetivos de negocio y mejora continua.

2.2. Métodos y procedimientos

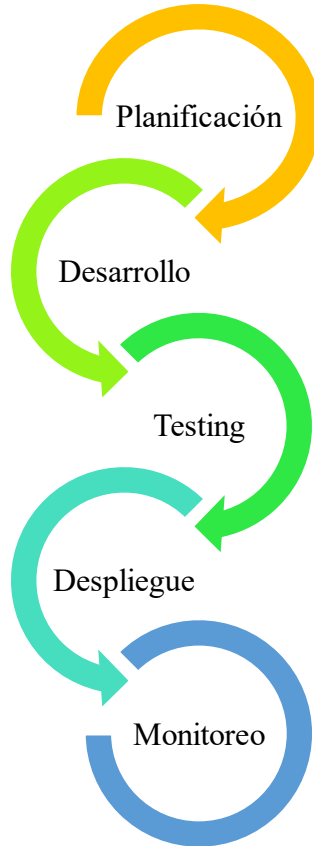
2.2.1. DevOps Lifecycle

Con el objetivo de mejorar el producto, existe lo conocido como el ciclo de vida DevOps, el cual se describe como la secuencia continua de actividades que integran el desarrollo, las operaciones y la entrega de *software*. Este ciclo incluye fases de planificación, desarrollo, pruebas, despliegue y monitoreo (Freixas, 2022).

A continuación, se detallan las fases del DevOps Lifecycle, así como en la **Figura 4** se visualiza la estructura que siguen estas fases:

1. **Planificación:** definición de requerimientos, priorización de funcionalidades y alineación con los objetivos técnicos del negocio.
2. **Desarrollo:** se desarrollan los requerimientos o *features* de *software*, generalmente cortos y colaborativos.
3. **Pruebas:** se automatizan las pruebas unitarias, funcionales y de integración para la validación de la calidad del código.
4. **Despliegue:** se entrega o despliega el *software* en el entorno de producción, de forma automatizada y con despliegues constantes.
5. **Monitoreo:** se evalúa el rendimiento del sistema en producción mediante métricas clave que sean de beneficio para la mejora continua.

Figura 4: DevOps Lifecycle



Fuente: elaboración propia.

El ciclo de vida de DevOps no es precisamente técnico en todos sus aspectos, cuenta con procedimientos culturales y de procesos de gestión de proyectos. Este enfoque se fundamenta dentro de tres principios claves, denominados *The Three Ways*, descritos específicamente en el libro *The DevOps Handbook*. A continuación, se detallan las formas de pensamiento y organización para lograr correctamente el DevOps Lifecycle:

1. **Primera forma (*First Way*):** busca facilitar el flujo continuo de trabajo desde el desarrollo hasta la etapa operativa del proyecto ya establecido en producción.
2. **Segunda forma (*Second Way*):** mediante la gestión de mejora continua y constante, se promueve la retroalimentación constante y rápida en todas las etapas del flujo de valor, permitiendo detectar errores de forma temprana.
3. **Tercera forma (*Third Way*):** se fomenta la cultura generativa, donde se apoya la toma de riesgos dinámicos necesarios para la innovación y la adaptación al cambio.

2.2.2. CI/CD Pipeline

La metodología DevOps se ve caracterizada por la integración y despliegue continuo de soluciones, acelerando la transición entre fases del ciclo de vida de un proyecto de *software*. *Continuous Integration / Continuous Deployment (CI/CD)* permite a los equipos entregar *software* de forma rápida, segura y frecuente.

Para lograr esto, se construye un *pipeline* de CI/CD, el cual consiste en una secuencia efectiva y automatizada de pasos que facilita la integración de código, la ejecución de pruebas unitarias y despliegue en entornos de producción. Tal como señalan Humble y Farley, esta práctica desarrolla sobre la disciplina de mejora continua, integración de compilación, pruebas unitarias y validaciones constantes. Además, según *The DevOps Handbook*: “El *pipeline* consiste en un mecanismo para asegurar que el código como la infraestructura siempre se encuentren en un estado de despliegue, y que de igual forma todo se vea integrado en la rama principal (*trunk*)” (Kim et., 2021).

Se puede asegurar que el manejo de un *pipeline* puede incluir fases en su desarrollo sumamente relacionadas a la conducta DevOps. A continuación, se desarrollan las etapas de este ciclo

- a. **Commit:** cada cambio de código se registra en el repositorio mediante el sistema de control de versiones, en este caso, **Git**.
- b. **Desarrollo:** el código es compilado y procesado para su verificación.
- c. **Test:** se ejecutan pruebas automatizadas.
- d. **Despliegue:** el sistema se despliega en entorno de producción.
- e. **Monitoreo:** se recopilan métricas para evaluar la versión que fue desplegada en producción.

El *pipeline* tiene como objetivo validar automáticamente cada cambio realizado en el código, asegurando su correcta integración en el ambiente de producción. En esta etapa se ejecutan pruebas automatizadas orientadas a garantizar tanto el rendimiento fluido del proyecto como la seguridad y precisión de las validaciones de código. Para ilustrar este proceso, a continuación, se presenta el flujo general de un *pipeline* de despliegue típico que incorpora estas etapas de verificación.

Figura 5: El pipeline de despliegue



Fuente: elaboración propia. Basada en “*The Deployment Pipeline*” de *The DevOps Handbook* (2021).

2.2.3. Pruebas automatizadas

Las pruebas automatizadas son una práctica fundamental dentro de la metodología DevOps y constituyen un pilar clave en las practicas CI/CD. Estas permiten validar de forma continua el funcionamiento del código fuente durante el ciclo de vida de desarrollo de los proyectos.

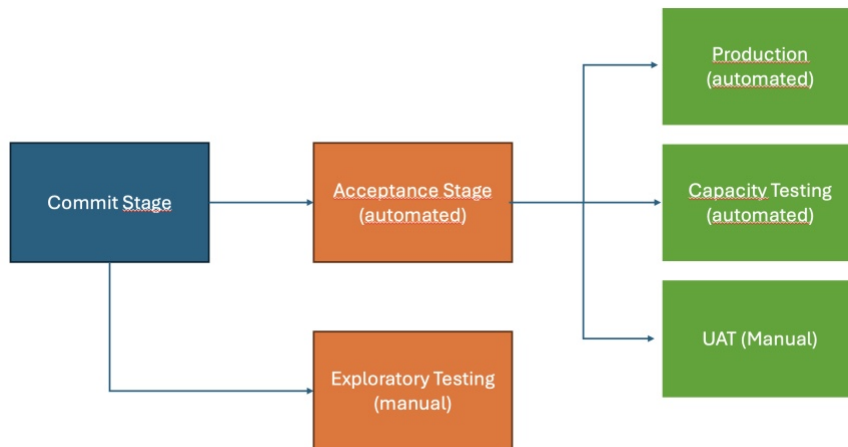
Como señalan los autores del libro *The DevOps Handbook*, “las pruebas automatizadas se ejecutan de forma recurrente cada vez que se integra un nuevo cambio en el control de versiones, permitiendo la validación inmediata del comportamiento del sistema en entornos similares a producción. Este mecanismo asegura que cualquier error en desarrollo, prueba o integración sea detectado” (Kim et., 2021).

La integración de estas pruebas dentro del pipeline tiene como objetivo detectar errores antes de se presenten en producción, manteniendo un flujo de validación constante. En este sentido, según *The DevOps Handbook*, “Primeramente, se crean suites de pruebas automatizadas que aumentan la frecuencia de integración y prueba del código, pasando de flujos periódicos a uno fijo continuo ” (Kim et., 2021). La incorporación de pruebas unitarias dentro de un pipeline automatizado es indispensable para lograr una retroalimentación eficaz y continua, siendo uno de los elementos esenciales de cualquier implementación DevOps.

Las pruebas automatizadas impulsan una cultura de mejora continua al facilitar la validación constante de los cambios, asegurando que estos cumplan con los criterios técnicos y de calidad definidos por la organización, así como los que son establecido en el AI Readiness Criteria.

A continuación, en la siguiente imagen se visualiza el flujo de las pruebas automatizadas dentro del pipeline CI/CD:

Figura 6: Flujo de pruebas automatizadas



Fuente: elaboración propia. Basada en “Running Automated and Manual Tests” de *The DevOps Handbook* (2021).

2.2.4. Code Review

El *code review* es una de las prácticas esenciales en el mundo de desarrollo de software, la cual se ha incorporado en múltiples marcos de trabajo para el cumplimiento de estándares en proyectos de índole informática.

Según el estudio titulado “*Automated Code Review in Practice*” se define *code review* como: “La práctica *Modern Code Review* caracterizada por ser informal, frecuente y asistida por herramientas colaborativas. Entre sus beneficios se destacan la detección de errores, promoción de buenas prácticas de código, la transmisión de conocimiento técnico y la mejora continua del producto” (Cihan et al., 2024). Este proceso implica que los desarrolladores revisen el código propuesto antes de su integración definitiva en producción, brindando sugerencias de mejora y posibles correcciones sobre errores dentro del procedimiento.

Dentro del marco DevOps, es sumamente importante la integración correcta de *code review* ya que cumple una pauta en la promoción de buenas prácticas de programación, minimización de errores y mejora continua a través de la integración cambios constantes.

2.2.5. Evaluación de madurez de procesos con prácticas DevOps

En las empresas que buscan una digitalización, modernización o la integración de metodologías de *software* efectivas, es necesario aplicar la evaluación de madurez de procesos. En referencia a DevOps y la ingeniería de inteligencia artificial, este diagnóstico es clave para identificar brechas, oportunidades de mejora y asegurar una progresión continua en marcos de trabajo más eficientes.

Para aplicar la evaluación de madurez de procesos, es necesario contar con distintas métricas que validen el trabajo que se realiza continuamente por un equipo. Según Forsgren en *Accelerate*, existen diferentes medidas de desempeño aplicables al despliegue de *software*, sin embargo, DevOps se ve beneficiado del uso del concepto de *Lead Time* (Forsgren et al., 2018).

2.2.5.1. Lead Time

Hace referencia al tiempo que transcurre desde que se realiza un cambio en el código hasta que se despliega en producción. Esta métrica resulta esencial dentro del pensamiento Lean, ya que mide el tiempo desde que se realiza una solicitud hasta que se entrega el valor.

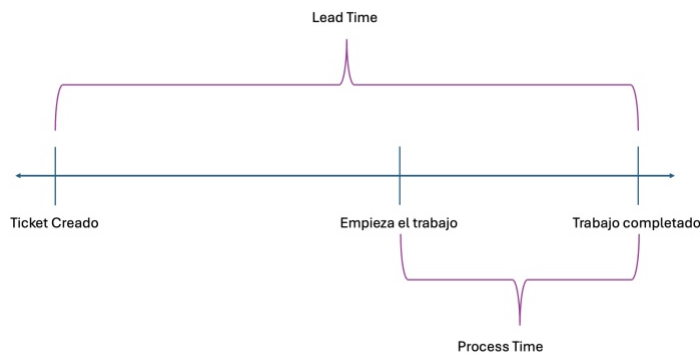
Según *The DevOps Handbook*, el flujo de valor empieza desde que se realiza cualquier cambio dentro del rango de Desarrollo, *QA*, operaciones de TI e Infosec, resultando en un cambio del control de versiones, hasta ser desplegado dentro de

producción (Kim et., 2025). Este rango de tiempo define el valor que puede ser entregado desde que se crea el requerimiento a su implementación concreta.

Dentro del procedimiento de *Lead Time*, se encuentra el *Process Time*, el cual es el tiempo en proceso en que se ejecuta la funcionalidad para que pueda ser integrada en producción. Este tiempo no toma en cuenta todo el procedimiento que puede estar en el *backlog* esperando a ser desarrollado.

Para la representación correcta de lo que es el procedimiento y evaluación del *Lead Time*, se ilustra la siguiente figura:

Figura 7: *Lead Time versus Process time*



Fuente: adaptado de *The DevOps Handbook* (Kim et., 2025, p.9).

2.2.5.2. Frecuencia de despliegues

La medición de frecuencia de despliegues es una de las formas más efectivas para medir el rendimiento de un equipo en cuestión cuando se habla de la metodología DevOps. En el paradigma Lean, esto es medido en frecuencia de lotes, sin embargo, en *software*, la medición es realizada por los lanzamientos de *software* que se realizan a producción. Según el libro *Accelerate*: “Disminuir en tamaño de los lotes reduce los tiempos de ciclo, variabilidad en el flujo, acelera la retroalimentación, reduce el riesgo, mejora la eficiencia, reduce costos y el crecimiento del cronograma. En *software*, una versión normalmente consiste en múltiples *commits* al control de versiones, donde cada commit puede ser liberado directamente a producción, integrando la práctica conocida como despliegue continuo” (Forsgren et al., 2018).

2.2.5.3. Change Fail Rate

Esto es conocido como el porcentaje de cambios que pueden generar errores en producción. Esta métrica permite evaluar la estabilidad y fiabilidad del sistema ante nuevas implementaciones. Siguiendo la metodología Lean, esto puede verificarse como el porcentaje de entregas que son correctas y completas. Tal como se señala en *Accelerate*: “Una métrica clave al realizar los cambios en los sistemas es qué porcentaje de cambios en producción (incluyendo, por ejemplo, lanzamientos de *software* y cambios en la configuración de infraestructura) fallan. En el contexto de

lean, esto es equivalente al porcentaje completo y correcto del proceso de entrega del producto, el cual representa una métrica esencial de calidad” (Forsgren et al., 2018).

2.3. Herramientas, prácticas y tecnologías

2.3.1. Herramientas de CI/CD

En el ciclo DevOps, es necesario concretar la automatización, validación, integración y despliegue continuo del código. Por esta razón, es necesario integrar herramientas que cumplan con estas características principales, que aseguren la entrega constante de valor al negocio.

Estas herramientas permiten desarrollar el proceso desde que se realiza un *commit* hasta que este código sea validado y, posteriormente, integrado en producción. Según *The DevOps Handbook*, automatizar estos procesos no solamente reduce errores humanos, sino que aumenta la calidad y la velocidad de entrega (Kim et al., 2021). Asimismo, *Accelerate* destaca que las organizaciones de alto rendimiento utilizan herramientas enfocadas en CI/CD para mejorar métricas como el *Lead Time*, frecuencia de despliegue y el *change failure rate* (Forsgren et al., 2018).

En el contexto de este proyecto, y según las necesidades del equipo de Ingeniería de Inteligencia Artificial, se destacan las siguientes herramientas clave:

2.3.1.1. Databricks Asset Bundles

Databricks Asset Bundles se comprende de una herramienta diseñada para facilitar la adopción de buenas prácticas de ingeniería de *software*, como el control de versiones, revisión de código, *unit testing*, CI/CD, aplicadas a proyectos de datos e inteligencia artificial (Databricks, 2025).

Los *bundles* permiten desarrollar recursos de Databricks como *notebooks*, *pipelines*, *Jobs* y modelos, estructurados mediante una configuración YAML que permiten la integración del ciclo de CI/CD.

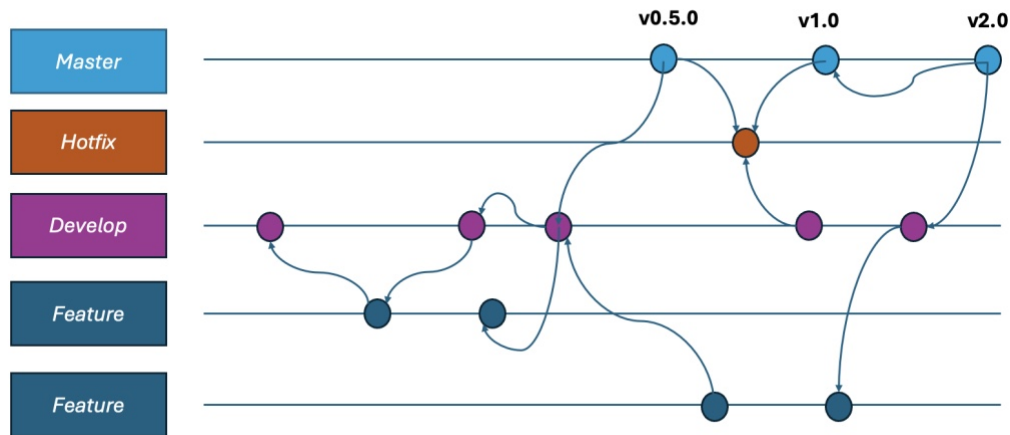
2.3.1.2. GitHub Actions

GitHub Actions es una plataforma de CI/CD que permite automatizar flujos de trabajo directamente desde el repositorio, facilitando el desarrollo, pruebas automatizadas y despliegue de *software* ágil mediante un *pipeline*. (GitHub, 2025). La adopción de GitHub Actions permite la gestión de DevOps centrado en el control de versiones, automatización y trazabilidad del código.

2.3.2. Git Flow

En entornos colaborativos, los sistemas de control de versiones, tales como Git, resuelven múltiples problemas del desarrollo de *software*, ya sea la pérdida de versiones estables, conflictos de *branches* y desorganización en los cambios aplicado en *features* o en alguna integración del código. Según Álvarez (2021), el uso del Git Flow implica estructurar el trabajo mediante distintas ramas o, en su defecto, *branches*, que evitan la integración de errores dentro de las versiones operacionalizables del código, asegurando la confiabilidad del producto. Además de permitir una trazabilidad de los cambios realizados, promueve una cultura de integración continua y responsabilidad sobre el desarrollo de *software*.

Figura 8: Git Flow Model



Fuente: elaboración propia. Basada en “Git Flow a succesful git branching model” de Álvarez Carulla, A. (2021).

Como se visualiza en la Figura 6, esta estrategia se basa en el desarrollo de *branches* que permiten registrar, organizar y gestionar los cambios en el código. Esta estructura se comprende de la siguiente forma:

- Master o main:** contiene versiones estables del código, definido como un estado de producción.
- Develop:** sirve como una rama de integración donde se agrupan todas las nuevas funcionalidades antes de ser desplegadas.
- Feature:** *Branches* destinadas a la creación de nuevas funcionalidades específicas.
- Hotfix:** ramas especializadas para corregir errores críticos directamente sobre la *main Branch*.

2.3.3. Herramientas de seguimiento

El seguimiento de la calidad de código es un concepto muy presente dentro del ciclo DevOps, este es esencial para garantizar la eficiencia y la mejora continua de proyectos de ingeniería de inteligencia artificial. En este caso, las herramientas de seguimiento son enfocadas en monitorear aspectos técnicos, organizacionales y de colaboración.

2.3.3.1. SonarQube

SonarQube es una herramienta de código abierto orientada al análisis efectivo del código fuente de un proyecto de *software*, buscando evaluar la calidad, la automatización de pruebas, seguridad y mantenibilidad del *software* durante su desarrollo. Esta herramienta está diseñada para integrarse en el ciclo de desarrollo y facilitar la identificación temprana de defectos y vulnerabilidades (Ospina Delgado, 2015).

2.3.3.2. Jira

Jira es una herramienta de gestión ágil de proyectos desarrollada por Atlassian, la cual busca planificar, monitorear y entregar *software* de forma estructurada y eficiente. En entornos DevOps, es ampliamente ejecutada, ya que permite flujos de trabajo por medio de objetivos de la organización. A su vez, Jira es adaptable a múltiples metodologías ágiles tales como Scrum y Kanban (Atlassian, 2024).

2.3.3.3. Git

Git es un sistema de control de versiones adoptado para la gestión de código fuente. El diseño de su arquitectura permite al desarrollador colaborar y desarrollar de forma funcional mediante un repositorio. A diferencia de otros sistemas de control de versiones git, permite el desarrollo mediante diferentes *branches*, seguimiento detallado de cambios, integración con CI/CD y soporte para flujos de trabajo avanzados (Microsoft, 2023).

2.3.4. Herramientas de documentación

La documentación es una práctica fundamental dentro de la metodología DevOps, es precisamente esencial para la estandarización de procedimientos y de entornos colaborativos, donde múltiples colaboradores pueden intervenir en procesos de desarrollo, validación y despliegue. Documentar puede ser la línea entre la integración correcta de CI/CD y, por ende, entregas veloces, ya que es una parte clave del flujo de valor de cada marco de trabajo.

2.3.4.1. Confluence

Confluence es una plataforma de trabajo colaborativo orientada a la documentación dinámica y estructurada de proyectos. Además, permite a los equipos crear, capturar y organizar toda la información clave para un proyecto, dando visibilidad del conocimiento de este y una coordinación efectiva entre miembros del equipo (Atlassian, 2024).

Además, Confluence es una herramienta clave para el desarrollo de un marco estructurado basado en DevOps, ya que permite desarrollar documentación técnica, operativa y metodológica de los proyectos, así como un lugar centralizado para definir los diferentes aspectos que pueden estandarizar un marco de trabajo para el equipo.

3. Marco Metodológico

3.1. Tipo de investigación

En esta sección, se desarrolla el tipo de investigación seleccionado para el Trabajo Final de Graduación, acompañado de una descripción de los tipos de investigación existentes.

La investigación es de tipo básica, esta busca desarrollar el conocimiento teórico, según Sarango et al. (2024): “La investigación básica o fundamental no tiene un enfoque inmediato en aplicaciones prácticas, sino que se centra en comprender los principios fundamentales”. Este tipo de investigación pretende incrementar los conocimientos integrados, permitiendo explicar, comparar y exponer los conceptos y términos fundamentados.

Además, según Sarango et al. (2024): “La investigación aplicada, se encuentra dirigida en la resolución de problemas específicos, aplicando los conocimientos en situaciones prácticas”. Por esta razón, la propuesta de un marco de trabajo estandarizado para el equipo de Ingeniería de Inteligencia Artificial busca generar un conocimiento previo, para, posteriormente, desarrollar y brindar una solución a la situación problemática.

3.2. Enfoque y diseño de la investigación

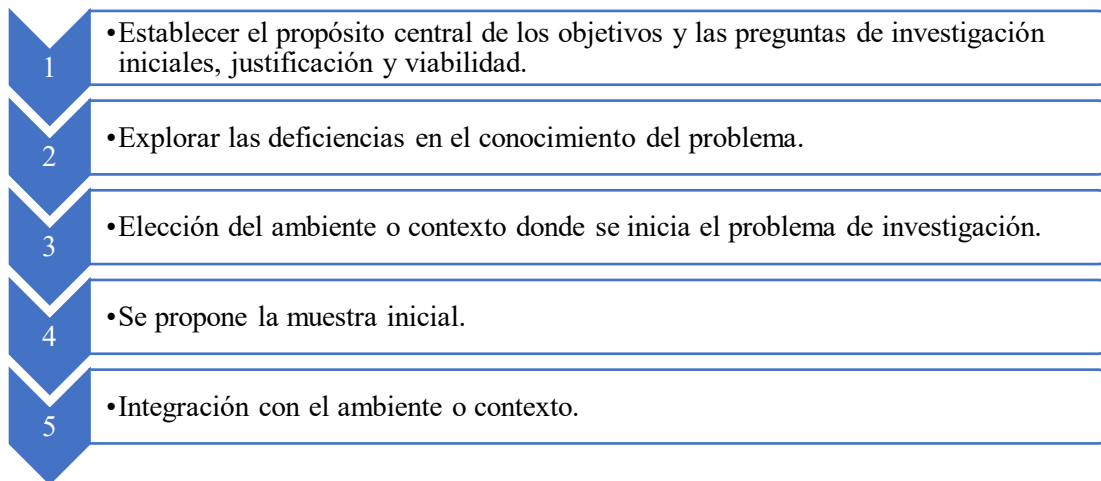
El enfoque de la investigación es mixto, ya que combina elementos cualitativos y cuantitativos, con el objetivo de comprender, analizar y proponer una solución viable. El componente cualitativo busca interpretar diferentes procedimientos que se presentan en el proyecto. Según Acosta Faneite (2023): “El enfoque cualitativo es de carácter subjetivo, que busca desarrollar la recolección de información abierta y permita una comprensión general del fenómeno, donde se busca comprender el propósito del análisis de los datos, mediante el contexto y el entorno”. Este enfoque permite visualizar el entorno de investigación, verificando el propósito u objetivo, la problemática y la justificación de la investigación. De esta forma, se pretende explorar las deficiencias de conocimiento, definiendo el contexto y la viabilidad del estudio.

Por otro lado, el enfoque cuantitativo se integra para medir métricas, comparar variables, eficiencia, frecuencia de errores, tiempos de entrega y adopción de herramientas dentro del ciclo de vida de los proyectos. Su finalidad es la búsqueda de evidencia que fortalece la propuesta realizada.

Según Hernández-Sampieri y Mendoza (2020): “La esencia de la investigación cualitativa radica en la comprensión de los fenómenos, explorándolos desde la perspectiva del entorno, ya sea los participantes y su relación con el contexto”. Este enfoque permite desarrollar un conocimiento que se basa en el entorno, en la investigación del funcionamiento de este, el cual se implementa para desarrollar la propuesta del marco de trabajo estandarizado para el equipo de Ingeniería de Inteligencia Artificial.

A continuación, se visualiza el proceso de investigación cualitativa, el cual se implementa mediante pasos:

Figura 3: Proceso de investigación cualitativa

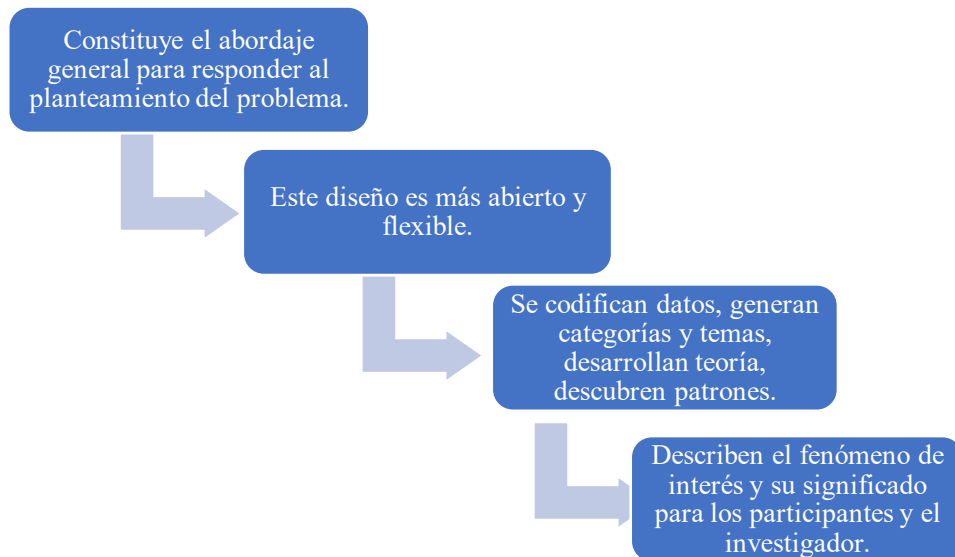


Fuente: se adapta de Hernández-Sampieri y Mendoza (2020).

El diseño cualitativo pretende reforzar sus directrices en el desarrollo de información, el cual se basa en la recolección y análisis de datos, para, posteriormente, concebir algún diseño básico que se integre en la investigación y, por ende, desenvuelva el diseño cualitativo.

La elección del diseño en la ruta cualitativa y sus características se reflejan en la siguiente figura:

Figura 4: Elección del diseño en la ruta cualitativa



Fuente: se adapta de Hernández-Sampieri y Mendoza (2020).

Según Hernández-Sampieri y Mendoza (2020): “En el enfoque cualitativo, el diseño se refiere al abordaje general que se implementa en el proceso de investigación. El diseño integra la recolección y el análisis, desarrollándose desde el planteamiento del problema hasta inmersión inicial para principalmente enfocar el fenómeno de interés”. De esta forma, se elige el criterio de investigación/acción, este diseño cualitativo busca resolver la problemática mediante fases cíclicas, identificando la problemática, su implementación, evaluación y finaliza con retroalimentación de dicha problemática. De esta forma, la investigación planteada para este proyecto desarrolla una perspectiva tanto práctica como participativa. Finalmente, se elige el diseño metodológico de tipo investigación-acción, analizando un problema y desarrollando una propuesta de solución mediante un PoC representativo.

3.3. Fuentes de datos e información

Para llevar a cabo el Trabajo Final de Graduación, se hace uso de las fuentes primarias y secundarias que son de importancia para el proyecto, justificando la realización de este. Además, en esta sección, se especifica por qué se utiliza dicha referencia.

3.3.1. Fuentes primarias

En la siguiente figura, se describen las fuentes primarias consultadas a lo largo del desarrollo de este proyecto.

Tabla 2: Fuentes primarias

Fuente primaria	Importancia
Sesiones, reuniones o entrevistas con el personal del equipo de Inteligencia Artificial	Sesiones para la obtención de información detallada de los procedimientos realizados por los integrantes del equipo de Ingeniería de Inteligencia Artificial para el desarrollo de los proyectos.
Documentación interna de la organización.	Análisis de los diferentes recursos con los que cuenta el equipo de Inteligencia Artificial a nivel de Unidad de Cuidado Familiar. Así como recursos, criterios o proyectos que se realizan en la organización de Inteligencia Artificial.
Documentación externa a la organización.	Análisis de proyectos, criterios y recursos externos que proporcionan información sobre el uso de metodologías ágiles y DevOps CI/CD.
Análisis directo del entorno del equipo de Ingeniería de Inteligencia Artificial.	Procedimiento realizado por el desarrollador del proyecto para identificar las deficiencias, limitaciones y oportunidades de mejora del equipo de Ingeniería de Inteligencia Artificial.

Fuente: Elaboración propia.

3.3.2. Fuentes secundarias

Según la siguiente cita, las fuentes secundarias se describen como: “El resultado del análisis documental de descripciones, resúmenes, bibliografía, entre otras fuentes, en las cuáles una persona ha trabajado sobre el contenido de estas” (Cabrera Méndez, 2010).

Tabla 3: Fuentes secundarias

Fuente secundaria	Importancia
Forsgren, N., Humble, J. y Kim, G. (2018). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations (1.a ed.). IT Revolution.	Este libro es clave para la conceptualización del Trabajo Final de Graduación, ya que ofrece evidencia sobre las prácticas y capacidades que caracterizan a distintas organizaciones tecnológicas en prácticas de alto desempeño. A través de análisis estadísticos, se identifica la adopción de prácticas como la integración continua, entrega continua y los despliegues automatizados.

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Fuente secundaria	Importancia
González, G. N. C., y Calderón, J. A. F. (2023). Integración y despliegue continuo con DevOps como cultura en empresas del sector TI colombianas.	Esta fuente secundaria es un artículo de revisión, el cual analiza las ventajas, desafíos y herramientas de integración de DevOps CI/CD en el contexto empresarial de Colombia. Se destaca por brindar información del ciclo de desarrollo tradicional, y cómo la transformación con DevOps promueve procesos más ágiles y colaborativos. Este artículo proporciona una base sólida en la propuesta de integración de CI/CD y su impacto en la eficiencia y calidad del desarrollo de proyectos.
Kim, G., Humble, J., Debois, P., y Willis, J. (2021). <i>The DevOps Handbook: How to create world-class Agility, Reliability & Security in technology organizations</i> (2.ª ed.).	Este libro es un manual ampliamente reconocido y premiado en el ámbito de la transformación digital y la implementación de la metodología DevOps. Además, integra los conceptos básicos de DevOps hasta los marcos de trabajo de esta metodología, los cuales incluyen el flujo, la retroalimentación y el aprendizaje continuo.
Mamani Larico, B. N. (2020). Implementación de ambientes de integración y despliegue continuo (CI/CD)	Esta fuente ha sido realizada en la institución UPeU Campus Juliaca, en la cual se analiza un modelo de la administración de aplicaciones empresariales mediante la metodología CI/CD. Este estudio aborda la problemática de la ausencia de estandarización en procesos de despliegue, proponiendo soluciones basadas en la mejora continua enfocándose en entregas de mayor calidad.
Mendoza Torres, A. (2023). <i>Integración del proceso de diseño de User Experience al ciclo DevOps CI/CD</i>	Se toma como referencia la integración del ciclo DevOps y CICD para el diseño de <i>User Experience</i> . Se destaca la finalidad de entregar <i>software</i> de calidad, lo que proporciona un insumo de valor para este proyecto, agregando insumos, los cuales reflejan la aplicabilidad del ciclo DevOps y CICD.
Project Management Institute. (2021). <i>PMBOK</i> (Guide 7th edition).	Este documento proporciona los principios esenciales para la gestión de proyectos, los cuales son aplicables para las necesidades de estandarización y mejora de procesos en el equipo de ingeniería de AI.
Tesis y proyectos finales de graduación de la Escuela de Administración de Tecnología de Información.	Estas referencias mencionadas en la sección de trabajos similares son insumos que abordan aspectos referentes a la gestión de proyectos, los cuales brindan información sobre propuestas, metodologías e ideas referentes al tema en cuestión.

Fuente: elaboración propia.

3.4. Sujetos de investigación

A continuación, se evalúan los distintos sujetos de investigación en torno a este proyecto. Según Rodríguez y Rodríguez (2020): “El sujeto de investigación es el individuo que asume el papel de investigador, es quien adentra el conocimiento y comprensión del objeto en investigación, esto con realización al estudio del objeto de investigación”.

Para este estudio, se contempla la participación de tres colaboradores de la organización de Ingeniería de Inteligencia Artificial. Cabe recalcar que estas intervenciones en formato de entrevista son con consentimiento informado, con el fin de garantizar la comprensión del propósito del estudio, respetando los principios éticos de la organización y del proyecto.

Tabla 4: Sujetos de investigación

Rol del sujeto	Años de experiencia en el rol	Caracterización del sujeto (diferentes responsabilidades y funciones del rol)	Justificación de la importancia de este sujeto para su investigación.
<i>Senior AI Engineer</i> del área de cuidado Familiar.	Cuatro años	Responsable de supervisar y gestionar proyectos de Ingeniería de Inteligencia Artificial en los ámbitos de manufactura y comercial de la Unidad de Cuidado Familiar. Coordina las actividades de dos AI Engineers y un pasante de AI Engineering, asegurando el cumplimiento de objetivos técnico y de negocio.	<ul style="list-style-type: none">• Principal conexión de comunicación con el equipo proveedor gestión de datos en Databricks.• Administradora de proyectos a nivel de manufactura y área comercial, los cuales son los principales enfoques de la unidad de cuidado familia.• Este sujeto es clave para evaluar la efectividad tanto técnica como operativa de la implementación correcta de DevOps y CI/CD en el equipo.

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Rol del sujeto	Años de experiencia en el rol	Caracterización del sujeto (diferentes responsabilidades y funciones del rol)	Justificación de la importancia de este sujeto para su investigación.
<i>AI Engineer</i> en el área de SRE	Dos años	<i>AI Engineer</i> enfocado en Site Reliability Engineering (SRE), el cual es el responsable de supervisar herramientas de automatización, monitorear incidentes y garantizar la estabilidad de las soluciones implementadas en producción.	<ul style="list-style-type: none"> • “<i>Site Reliability Engineering</i>”, es parte del flujo DevOps, por lo que es de suma importancia visualizar el flujo actual del equipo de esta práctica. • Este sujeto es clave para medir la efectividad y la estabilidad de proyectos en producción del equipo de Ingeniería de Inteligencia Artificial. • Cabe recalcar que, en este momento, existen dos <i>SRE Engineers</i> en el equipo de AI Engineering.
<i>Senior AI Engineer</i> del área de Cuidado Familiar.	Dos años	Responsable de la gestión de proyectos de Ingeniería de Inteligencia Artificial en el área de manufactura de Cuidado Familiar. El enfoque de este profesional se centra en la ingeniería de datos y la ciencia de datos.	<ul style="list-style-type: none"> • Experiencia directa en la gestión de proyectos de ingeniería de inteligencia artificial dentro de la unidad de Cuidado Familiar. • Su enfoque técnico permite aportar perspectiva crítica sobre las necesidades reales del equipo, así como las oportunidades de mejora en la aplicación de un marco de trabajo estructurado.

Fuente: elaboración propia.

3.5. Variables o categorías de la investigación

En esta sección, se desarrollan las variables de la investigación, las cuales se emplean para evaluar el adecuado cumplimiento de los objetivos específicos previamente planteados. Asimismo, estas permiten identificar los instrumentos de utilidad para abordar las incógnitas que deben resolverse.

Tabla 5: Variables de investigación del objetivo 1

Objetivo específico N.º 1			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V01: Situación actual de la gestión de proyectos del equipo de Ingeniería de Inteligencia Artificial.</p> <p>Tipo: Dependiente.</p>	Estado actual de los procesos, herramientas y flujos de trabajo empleados por el equipo de Ingeniería de Inteligencia Artificial.	<ul style="list-style-type: none"> ○ Procedimientos y metodologías actuales del desarrollo de proyectos E2E del equipo. ○ Descripción del nivel de documentación en cada fase del ciclo de vida. ○ Grado de automatización en tareas clave. ○ Grado de deuda técnica actual del equipo. 	<ul style="list-style-type: none"> ○ <u>Entrevistas semiestructuradas</u> al equipo de Ingeniería de Inteligencia Artificial. Entrevista ○ <u>Revisión documental</u> de proyectos actuales. ○ <u>Diagrama Ishikawa</u> para identificación de causas.

Objetivo específico N.º 1

Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V02: Cumplimiento de métricas y estándares de calidad de proyectos de Ingeniería de Inteligencia Artificial.</p>	<p>Se describe como el nivel de cumplimiento de las métricas de calidad establecidas por la industria y la organización (AI Readiness Criteria) para las soluciones informáticas del equipo.</p>	<ul style="list-style-type: none"> ○ Métricas de cumplimiento de estándares de Ingeniería de Inteligencia Artificial. ○ Porcentaje actual de cumplimiento de métricas en los proyectos del equipo. ○ Calidad del código fuente. ○ Integración con tecnologías de confiabilidad, seguridad y sostenibilidad de proyectos de Ingeniería de Inteligencia Artificial. ○ Limitaciones técnicas y Operativas del equipo. 	<ul style="list-style-type: none"> ○ <u>Observación directa</u> de actividades para evaluar cumplimiento normativo ○ <u>Revisión documental</u> de las métricas de calidad. (AI Readiness Criteria) ○ <u>Entrevista semiestructurada</u> sobre limitaciones técnicas y operativas.

Fuente: elaboración propia.

Tabla 6: Variables de investigación objetivo 2

Objetivo específico N.º 2			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V03: Mejores prácticas de gestión de proyectos para el equipo de Ingeniería de IA</p> <p>Tipo: Independiente</p>	<p>Conjunto de procedimientos que aseguran la calidad y el éxito de los proyectos de Ingeniería de Inteligencia Artificial, alienados con metodologías ágiles y DevOps.</p>	<ul style="list-style-type: none"> • Herramientas utilizadas por el equipo para el desarrollo de soluciones informáticas. • Descripción detallada de roles, responsabilidades y metodologías aplicadas por el equipo. <ul style="list-style-type: none"> ○ Grado de adopción de prácticas de automatización en el equipo. ○ Uso de herramientas colaborativas para la gestión de proyectos. 	<ul style="list-style-type: none"> • <u>Revisión documental</u> de marcos de referencia. • <u>Revisión documental</u> y técnica de las mejores herramientas aplicables en el ciclo DevOps. • <u>Entrevista semiestructurada</u> sobre adopción de metodologías ágiles.

Objetivo específico N.º 2			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V04: Estándares operativos y técnicos del marco de trabajo DevOps CI/CD.</p> <p>Tipo: Dependiente.</p>	<p>Conjunto de procesos técnicos, operativos y administrativos que garantizan la eficiencia, calidad y estabilidad de los proyectos mediante la adopción del marco DevOps CI/CD. Estos procedimientos abarcan desde la planificación hasta el despliegue de soluciones informáticas.</p>	<ul style="list-style-type: none"> • Procesos documentados para el ciclo de vida del <i>software</i>. • Documentación de estándares definidos para garantizar la automatización de procesos en el ciclo DevOps. • Herramientas para el desarrollo de <i>pipelines</i> de CI/CD. 	<ul style="list-style-type: none"> • <u>Observación directa</u> del desarrollo para identificar estándares aplicados. • <u>Modelado de procesos BPMN</u> del flujo de trabajo en el equipo.

Fuente: elaboración propia.

Tabla 7: Variables de investigación objetivo 3

Objetivo específico N.º 3			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V05: Diferencias operacionales entre la situación actual y la propuesta de estandarización.</p> <p>Tipo: Independiente</p>	<p>Se refiere a las diferencias visualizadas en los procesos, herramientas y flujos de trabajo actuales del equipo de Ingeniería de Inteligencia Artificial, en comparación con la propuesta de estandarización del marco de trabajo.</p>	<ul style="list-style-type: none"> • Procesos modificados o agregados al estándar del marco de trabajo. • Herramientas y procedimientos integrados en el <i>pipeline</i> de desarrollo. • Beneficios operacionales identificados (reducción de errores, eficiencia en pruebas, velocidad de despliegue). 	<ul style="list-style-type: none"> • <u>Evaluación comparativa</u> de métricas con el AI Readiness Criteria.

Objetivo específico N.º 3			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V06: Mejoras en el ciclo de vida de los proyectos</p> <p>Tipo: Dependiente</p>	<p>Mejoras identificadas en eficiencia, calidad, estabilidad y cumplimiento de métricas tras el desarrollo de la propuesta de estandarización del marco de trabajo basado en DevOps CI/CD.</p>	<ul style="list-style-type: none"> • Descripción del ciclo de vida de proyectos abarcando planificación a despliegue de este. • Reducción en el tiempo de ejecución por cada etapa del ciclo de vida del proyecto. • Mejoras en las métricas de CI/CD. 	<ul style="list-style-type: none"> • <u>Revisión de herramientas y practicas</u> técnicas como operativas

Fuente: elaboración propia

Tabla 8: Variables de investigación objetivo 4

Objetivo específico N.º 4			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V07: Desarrollo de la prueba de concepto en un proyecto representativo de la Unidad de Cuidado Familiar.</p> <p>Tipo: Independiente.</p>	<p>Aplicación práctica del marco de trabajo estandarizado basado en DevOps CI/CD en un proyecto real, con el fin de validar su viabilidad, adaptabilidad y efectividad.</p>	<ul style="list-style-type: none"> • Grado de implementación de prácticas DevOps (control de versiones, automatización, integración y despliegue continuo) • Funcionalidades de CI/CD aplicadas en el <i>pipeline</i>. • Niveles de automatización alcanzados (<i>testing</i>, despliegue, tiempos de respuesta). 	<ul style="list-style-type: none"> • <u>Prueba de concepto</u> de actividades del proyecto (tareas, etapas, incidencias resueltas) • Evaluación técnica del <i>pipeline</i>. • Documentación del proceso y nivel de automatización alcanzado.

Objetivo específico N.º 4			
Nombre de la variable	Definición conceptual	Indicador	Definición instrumental
<p>V08: Evaluar y ajustar la propuesta de estandarización del marco de trabajo.</p> <p>Tipo: Dependiente.</p>	<p>Evaluación de los cambios realizados en función de la prueba de concepto, para validar y ajustar la efectividad de esta misma.</p>	<ul style="list-style-type: none"> Nivel de adecuación y adaptación del marco de trabajo estandarizado a las condiciones reales del equipo de la Unidad de Cuidado Familiar. Retroalimentación del equipo sobre la utilidad y aplicabilidad del marco de trabajo. 	<ul style="list-style-type: none"> <u>Comparación entre propuesta teórica y ejecución real.</u> Observación directa de la efectividad de la prueba de concepto. <u>Entrevistas semiestructurada con AI Engineer</u> para evaluar aplicabilidad y posibles ajustes.

Fuente: elaboración propia.

3.6. Técnicas e instrumentos de recolección de datos

En la siguiente sección, se describen las técnicas, herramientas o instrumentos de recolección de datos para la investigación. El propósito de estos recursos es recolectar la información para la gestión del mismo proyecto que, en este caso, corresponde a proyectos de Ingeniería de Inteligencia Artificial y la viabilidad, así como ajuste de la propuesta de estandarización basada en la metodología DevOps CI/CD.

Según Hernández Sampieri: “En la investigación cualitativa, la recolección de datos es un proceso fundamental que ocurre en ambientes naturales y cotidianos de los participantes y se orienta en obtener datos de personas, comunidades, situaciones o procesos en profundidad”. (Hernández Sampieri et al., 2014, p.397). Esto hace un enfoque en el investigador, el cual utiliza diversas técnicas para comprender el fenómeno estudiado.

A continuación, se describen las técnicas e instrumentos aplicados en este proyecto:

3.6.1. Revisión documental

Esta herramienta tiene como funcionalidad principal ser el contexto de estudio, y, por ende, fundamentar teóricamente la propuesta de estandarización. Según Hernández Sampieri: “La revisión de la literatura consiste en detectar, consultar y obtener bibliografía y otros materiales útiles para propósitos de estudio, de los cuáles se extrae y recopila información relevante para el problema de investigación” (Hernández Sampieri et al., 2014, p.53). De tal forma, la adquisición de conocimientos técnicos y operativos para el desarrollo de la propuesta del marco de trabajo

estandarizado permite abordar situaciones actuales de gestión de proyectos, estándares y prácticas que son fundamentadas mediante la revisión bibliográfica.

3.6.2. Entrevista Semiestructurada

Las entrevistas son uno de los principales métodos para obtener información cualitativa, representando la experiencia directa de los miembros del equipo de Ingeniería de Inteligencia Artificial. Según Hernández Sampieri: “La entrevista se define como una reunión para el intercambio de información entre una persona (el entrevistador) y otra (el entrevistado).” (Hernández Sampieri et al., 2014, p.403). El insumo de entrevista tiene un rol fundamental para la adquisición de conocimiento influido por líderes técnicos, desarrolladores y el equipo de Ingeniería de Inteligencia Artificial.

En esta investigación, se emplea la modalidad de entrevista semiestructurada, la cual permite al entrevistador introducir preguntas adicionales para profundizar o aclarar en conceptos relevantes. Este tipo de entrevista resulta especialmente útil para captar la perspectiva de los *AI Engineers* e identificar limitaciones y oportunidades de mejora dentro del ciclo de vida de proyectos.

3.6.3. Encuesta

Según Hernández Sampieri: “Las encuestas son investigaciones no experimentales transversales o transeccionales descriptivas o correlacionales-causales. Generalmente utilizan cuestionarios que se aplican diferentes contextos” (Hernández Sampieri et al., 2014, p.158). Las encuestas se encuentran dirigidas al equipo de Ingeniería de Inteligencia Artificial, permitiendo obtener percepciones sobre el impacto de la propuesta del marco de trabajo, nivel de automatización y satisfacción con los resultados del proyecto.

3.6.4. Observación directa

Según Hernández Sampieri: “Las anotaciones de la observación directa consisten en descripciones de lo que se ve y escucha según el contexto o los participantes en cuestión.” (Hernández Sampieri et al., 2014, p.377). La observación directa es fundamental para la identificación de limitaciones, estándares y oportunidades de mejora aplicables dentro del marco de estandarización propuesto.

3.6.5. Modelado de procesos BPMN

Según Windgassen: “BPMN permite a las organizaciones visualizar y optimizar sus procesos de negocio mediante una notación gráfica consistente, que facilita la comprensión y la comunicación transversal de los flujos de trabajo” (Windgassen, 2025). Esta técnica permite la representación comprensible del flujo de trabajo con la propuesta del marco de estandarización, verificando aspectos fundamentales de la metodología DevOps CI/CD.

3.6.6. Prueba de concepto

Según Acosta Faneite: “El significado de prueba de concepto es demostrar la viabilidad de un producto, método o idea. Corresponde a una forma de demostrar que la idea funcionará en la práctica” (Acosta Faneite, 2023). Con la prueba de concepto, se ejecutan actividades clave en el ciclo DevOps CI/CD, permitiendo validar tanto los resultados operativos como el nivel de ajuste de la propuesta.

3.7. Procedimiento metodológico de la investigación

En esta sección, se identifican los pasos específicos para llevar a cabo el proceso de investigación, donde se verifica la ejecución de las tareas de cada fase del procedimiento metodológico que permitan alcanzar los objetivos específicos propuestos anteriormente.

3.7.1. Diagrama propuesto para las fases del procedimiento metodológico

A continuación, se describen las fases correspondientes al procedimiento metodológico de la investigación, las cuales identifican las actividades necesarias para llevar a cabo la propuesta del marco de trabajo.

Tabla 9: Descripción de las fases del procedimiento metodológico

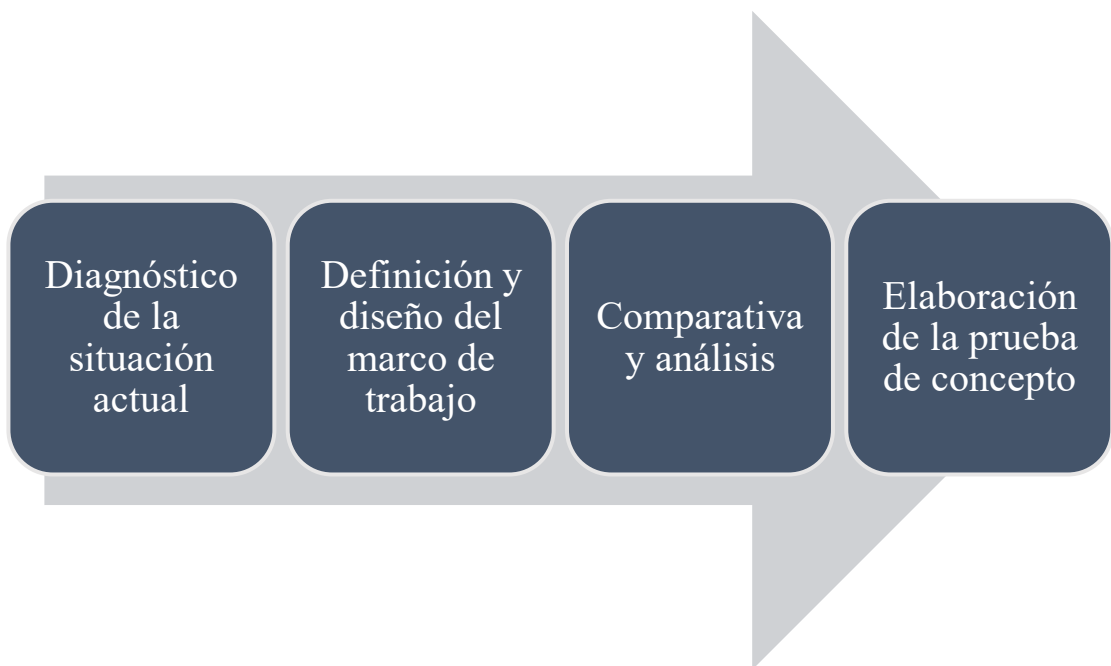
Fase	Descripción de actividades
1. Diagnóstico y análisis de la situación actual.	<p>La primera fase busca identificar y analizar las limitaciones actuales que presenta el equipo de Inteligencia Artificial para alinear los proyectos desarrollados a las métricas y prácticas de calidad de la organización.</p> <p>Para el desarrollo correcto de esta fase, se ejecutan las siguientes tareas:</p> <ul style="list-style-type: none">• Sesiones, entrevistas y reuniones exploratorias: se recopila información por medio de los miembros clave del equipo de Inteligencia Artificial, relacionado a identificar problemáticas y limitaciones en el desarrollo de las soluciones informáticas.• Análisis de herramientas y metodologías actuales: se verifican las plataformas actualmente utilizadas por el equipo para la gestión de proyectos, por ejemplo, Databricks y la gestión de datos para el equipo.• Análisis cualitativo: identificación de los cuellos de botella actuales en los procesos del ciclo de vida de los proyectos.• Aplicación del diagrama Ishikawa: instrumento utilizado como técnica de análisis para esclarecer las causas raíz de las problemáticas detectadas.

Fase	Descripción de actividades
<p>2. Definición y diseño del marco de trabajo</p>	<p>En la segunda fase, se busca definir los procesos operativos, técnicos y administrativos necesarios para el desarrollo de un marco funcional basado en la metodología DevOps CI/CD. Para el desarrollo correcto de esta fase, se ejecutan las siguientes taras:</p> <ul style="list-style-type: none"> • Identificación de las mejores prácticas en la industria: se investigan y seleccionan las mejores prácticas en relación con la metodología DevOps CI/CD. <p>Elección de procedimientos y herramientas: se valoran las herramientas y procesos que sean integrables al flujo de trabajo del equipo de Inteligencia Artificial.</p>
<p>3. Comparación y análisis entre el marco de trabajo anterior y el marco propuesto</p>	<p>La tercera fase tiene como propósito comparar las principales diferencias operacionales existentes entre la situación actual del equipo y la propuesta de estandarización. Para el desarrollo correcto de esta fase, se ejecutan las siguientes tareas:</p> <ul style="list-style-type: none"> • Análisis de puntos clave: identificación de cuáles son los procedimientos agregados o modificados que son de beneficio para el flujo de trabajo del equipo. • Análisis cualitativo: evaluar cada paso del ciclo de desarrollo con base en su impacto en la gestión de proyectos. <p>Recomendaciones: se desarrolla un documento comparativo que detalle los beneficios de la propuesta en comparación con el estado actual, identificando áreas de mejora, prácticas y ajustes necesarios.</p>

Fase	Descripción de actividades
<p>4. Elaboración de la prueba de concepto.</p>	<p>La cuarta fase consiste en operacionalizar la propuesta mediante una prueba de concepto en un proyecto representativo de la Unidad de negocio de Cuidado Familiar. En esta fase, se desea encontrar la viabilidad y los beneficios cuantitativos del marco de trabajo propuesto.</p> <ul style="list-style-type: none"> • Documentación de procesos y resultados: se recopilan las actividades realizadas durante la operacionalización, ya sean procedimientos, herramientas o resultados obtenidos. • Presentación de la metodología: elaboración del flujo de trabajo de la prueba de concepto a nivel de CI/CD con el uso de las herramientas previamente identificadas. Concluye con el flujo funcional y representativo de operacionalización del proyecto. • Resultados obtenidos de la prueba de concepto: se presentan los resultados obtenidos en el ciclo de vida del proyecto, abarcando desde la planificación hasta el despliegue. Se busca evaluar, mediante eficiencia, las métricas que administra la organización de Ingeniería de IA.

Fuente: elaboración propia.

Figura 9: Diagrama del procedimiento metodológico



Fuente: elaboración propia.

3.8. Operacionalización de las variables

En este apartado, se realiza la operacionalización de las variables de la investigación, el cual es el proceso de transformación de conceptos teóricos en elementos observables y medibles. La operacionalización garantiza que las variables se encuentren alineadas con los objetivos específicos del proyecto, y, por ende, las fases de este.

Tabla 10: Operacionalización de variables

Fase	Objetivo específico	Instrumentos	Variables	Sujetos de investigación
Diagnóstico y análisis de la situación actual.	Objetivo N.º 1	<ul style="list-style-type: none"> Entrevistas al equipo de Ingeniería de Inteligencia Artificial. Revisión documental de proyectos. Encuestas sobre calidad y eficiencia. 	<p>V01: Situación actual de la gestión de proyectos del equipo.</p> <p>V02: Cumplimiento de métricas y estándares de calidad.</p>	<ul style="list-style-type: none"> Senior AI Engineer SRE Engineer
Definición y diseño del marco de trabajo	Objetivo N.º 2	<ul style="list-style-type: none"> Entrevistas al equipo de Ingeniería de Inteligencia Artificial. Revisión documental de proyectos Encuestas sobre calidad y eficiencia. Modelado de procesos (BPMN). 	<p>V03: Mejores prácticas de gestión de proyectos.</p> <p>V04: Estándares operativos y técnicos del marco DevOps CI/CD.</p>	<ul style="list-style-type: none"> Senior AI Engineer (Enfoque en Data Engineering) Senior AI Engineer (Enfoque en Data Science) Senior AI Engineer (Enfoque AI Factory) SRE Engineer

Fase	Objetivo específico	Instrumentos	Variables	Sujetos de investigación
Comparación y análisis entre el marco de trabajo anterior y el marco propuesto	Objetivo N.º 3	<ul style="list-style-type: none"> • Observación directa. • Evaluación comparativa de Métricas (<i>AI Readiness Criteria</i>). 	<p>V05: Diferencias operacionales</p> <p>V06: Mejoras en el ciclo de vida de los proyectos.</p>	<ul style="list-style-type: none"> • <i>Senior AI Engineer</i>
Implementación de la prueba de concepto.	Objetivo N.º 4	<ul style="list-style-type: none"> • Evaluación técnica del <i>pipeline</i>. • Revisión de documentación y ejecución. • Entrevistas de retroalimentación. • Prueba de concepto. 	<p>V07: Desarrollo de la prueba de concepto.</p> <p>V08: Evaluación y ajuste de la propuesta.</p>	<ul style="list-style-type: none"> • <i>Senior AI Engineer</i>

Fuente: elaboración propia.

3.9. Tabla resumen del procedimiento metodológico de la investigación

En esta sección, se desarrolla la tabla resumen del procedimiento metodológico, en la cual es posible verificar las etapas claves del proyecto y su alineamiento con el marco conceptual, análisis de resultados, propuesta de solución y conclusiones.

Tabla 11: Tabla resumen del procedimiento metodológico de la investigación

Objetivo	Marco Conceptual	Metodología	Análisis de resultados	Propuesta de Solución	Conclusiones	Recomendaciones
Analizar la situación actual del equipo de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar, con el fin de identificar los requerimientos técnicos y operativos necesarios para implementar correctamente la metodología DevOps CI/CD.	Sección 2.1	Sección 3.5 Sección 3.6 Sección 3.7.1 Sección 3.8	Apéndice F. Entrevista N 1 Apéndice G. Entrevista N 2 Análisis de la situación actual		Sección 6.1	Sección 7.1 Sección 7.2
Diseñar el marco de trabajo basado en la metodología DevOps CI/CD, según los lineamientos operativos y técnicos identificados, asegurando las mejores prácticas ágiles que sean adaptables al desarrollo de proyectos del equipo de Inteligencia Artificial.	Sección 2.1 Sección 2.2 Sección 2.3	Sección 3.5 Sección 3.6 Sección 3.7.1 Sección 3.8	Apéndice G. Entrevista N 2 Análisis de la situación actual Diseño del marco de trabajo		Sección 6.2	Sección 7.1 Sección 7.2

Objetivo	Marco Conceptual	Metodología	Análisis de resultados	Propuesta de Solución	Conclusiones	Recomendaciones
Comparar las diferencias operacionales en términos de eficiencia, tiempos de entrega y calidad de las soluciones informáticas de la gestión de proyectos de Inteligencia Artificial entre la situación actual y la propuesta de estandarización basada en DevOps CI/CD, con el propósito de validar las mejoras implementadas en el ciclo de vida de los proyectos.	<p>Sección 2.1</p> <p>Sección 2.2</p> <p>Sección 2.3</p>	<p>Sección 3.5</p> <p>Sección 3.6</p> <p>Sección 3.7.1</p> <p>Sección 3.8</p>	<p>Comparación de la situación actual vs. Marco Propuesto</p>		<p>Sección 6.3</p>	<p>Sección 7.1</p> <p>Sección 7.2</p>
Elaborar una prueba de concepto en un proyecto representativo de la Unidad de Cuidado Familiar, priorizando la integración continua, automatización de pruebas, despliegue de soluciones e integración de CI/CD, evaluando el ajuste de la propuesta de estandarización del marco de trabajo de gestión de proyectos.	<p>Sección 2.1</p> <p>Sección 2.2</p> <p>Sección 2.3</p>	<p>Sección 3.5</p> <p>Sección 3.6</p> <p>Sección 3.7.1</p> <p>Sección 3.8</p>		<p>Sección 5.3</p>	<p>Sección 6.4</p>	<p>Sección 7.1</p> <p>Sección 7.2</p>

Fuente: elaboración propia.

4. Análisis de Resultados

En el presente capítulo se exponen los datos y resultados obtenidos a partir de la aplicación de las fases descritas en el Marco Metodológico. Inicialmente, se analiza la situación actual del equipo de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar, con el fin de identificar las principales limitaciones y necesidades operativas.

Posteriormente, se determinan los requerimientos técnicos y operativos necesarios para sustentar el desarrollo de una propuesta de marco de trabajo basada en la metodología DevOps CI/CD. A partir de esta base, se realiza una comparación estructurada entre el estado actual de la gestión de proyectos y el marco propuesto, destacando los beneficios identificados en términos de eficiencia, automatización y cumplimiento de estándares.

Finalmente, se presenta una prueba de concepto que permite ejemplificar, desde una perspectiva técnica, cómo se desarrolla un proyecto con la incorporación de las prácticas planteadas dentro de la propuesta; de esta forma, se valida su aplicabilidad y alineación con los objetivos estratégicos del equipo de Ingeniería de Inteligencia Artificial.

4.1. Fase 1: Análisis de la situación actual

Primeramente, se presentan los resultados obtenidos del análisis cualitativo basado en entrevistas semiestructuradas y revisión documental, enfocadas en la evaluación de la situación actual del equipo de Ingeniería de Inteligencia Artificial, alineado con el objetivo específico uno.

4.1.1. Resultados de entrevistas semiestructuradas

4.1.1.1. Primera entrevista

La primera entrevista se aplica a un *AI Engineer* del equipo de Ingeniería de Inteligencia Artificial, el cual brinda sus servicios específicamente para el desarrollo de proyectos de la Unidad de Cuidado Familiar. La estructura de dicha entrevista puede visualizarse en el siguiente apéndice: Apéndice F. Entrevista N 1.

A continuación, se describen los resultados obtenidos de la entrevista:

1. Proceso de desarrollo (*E2E*)

- El ciclo inicia con la identificación del *use case* en conjunto con el cliente.
- Posteriormente, se define cuál es el alcance técnico, para definir el valor en costo del proyecto.
- El desarrollo del proyecto no es estandarizado, es según el criterio del ingeniero y del cliente. Lo cual quiere decir que no existe una estructura clara para la gestión de estos
- No cuenta con documentación formalizada para el manejo de proyectos.

2. Herramientas utilizadas

- La comunicación se desarrolla mediante herramientas como, Outlook, Microsoft Teams y Office.
- El uso de herramientas colaborativas especializadas (Jira, Confluence) no se utiliza sistemáticamente, salvo en proyectos específicos, por lo que no es un estándar para el desarrollo de estos.
- No se dispone de herramientas de seguimiento E2E.

3. Cuellos de botella identificados

- Se presentan principalmente durante la definición del valor del proyecto con el cliente.
- La ambigüedad en los requerimientos o expectativas iniciales genera retrasos.
- El entrevistado destacó la necesidad de mayor claridad y estructura en esta etapa inicial. Esto debido a que es el *AI Engineer* el que debe proponer una solución según el problema comentado, los requerimientos no claros o estructurados en la mayoría de los casos.

4. Metodologías y procedimientos

- No existe una metodología clara para el desarrollo de los proyectos. Únicamente se reciben requerimientos y se desarrollan para llegar al resultado, por lo que se opera con base en la experiencia profesional y confianza individual.
- Para métodos de exploración de datos y gestión de estos se utiliza Databricks junto al Launchpad integrado por el gestor de los datos.
- Ausencia de lineamientos internos sobre prácticas DevOps o flujos CI/CD.
- El entrevistado no considera que el Launchpad para la gestión de los datos sea una limitante dentro de sus procedimientos dentro de la empresa.

5. Documentación

- No se elabora documentación técnica formal ni sistematizada.
- Se producen informes o presentaciones orientadas al cliente o áreas de negocio.

6. Nivel de automatización

- Se realizan pruebas de integración de forma manual, sin integrarlos dentro de un *pipeline* automatizado.
- No realiza automatización de pruebas unitarias ni despliegues continuos.
- Ausencia de un ciclo formal de *CI/CD*.
- Ausencia de integración de Infosec y de AI Factory dentro del flujo de trabajo de proyectos.

7. Deuda técnica

- Presencia de deuda técnica acumulada en procesos de automatización y estandarización.

- El código no sigue prácticas de refactorización debido a que los requerimientos son constantes a lo largo de las semanas.
- No existe integración de desarrollo, *testing* y despliegue automatizado.
- Existen múltiples procesos manuales a nivel de Excel y utilización de datos.

8. Cumplimiento de AI Readiness Criteria

- El marco del *AI Readiness Criteria* no es aplicado formalmente.

4.1.1.2. Segunda entrevista

La segunda entrevista se aplica a un *AI Engineer* del equipo de Ingeniería de Inteligencia Artificial, el cual brinda sus servicios específicamente para el desarrollo de proyectos de la Unidad de Cuidado Familiar. La estructura de dicha entrevista puede visualizarse en el siguiente apéndice: Apéndice G. Entrevista N 2.

A continuación, se describe el análisis realizado según los resultados obtenidos de la entrevista:

1. Descripción del proceso de desarrollo de un proyecto (E2E).

- La entrevistada destaca que los proyectos inician cuando una unidad de negocio plantea una necesidad específica.
- El primer paso es la reunión con el usuario para entender el flujo actual, identificar cuáles son los procesos manuales y, de esta forma, obtener muestras de datos.
- Posteriormente, los ingenieros realizan una etapa de exploración en las bases de datos de la empresa para identificar la información y datos requeridos, de esta forma, se busca replicar el flujo manual actual y validar los resultados con el cliente.
- Una vez que el objetivo está claro, el proyecto transiciona de una prueba de concepto a un proyecto formal con una estructura del código dentro de los *frameworks* formales de la organización.

2. Herramientas utilizadas para planificación, documentación y seguimiento.

- Durante etapas iniciales, la planificación se realiza de manera informal a través de correos electrónicos y reuniones no documentadas. Esta etapa se considera clave para entender el problema y la gestión de requerimientos.
- Solo hasta que el proyecto es aprobado para su despliegue en producción, se formaliza el seguimiento utilizando herramientas como *Jira* para *sprints* y asignación de tareas, y *Confluence* para documentación técnica.
- Se considera que esta estructura mixta es inconsistente, ya que en proyectos exploratorios la documentación es escasa o inexistente.

3. Principales cuellos de botella.

- El mayor cuello de botella identificado es en la fase de exploración de los datos. Esto es debido a la falta de acceso a ciertas plataformas, por ende, a la información. A su vez, es necesaria la disponibilidad de expertos que puedan explicar el funcionamiento de estas plataformas, por ejemplo, el Launchpad que utiliza el equipo de Ingeniería de Inteligencia Artificial.

4. Uso de metodologías y flujos estandarizados.

- Tomando en cuenta el ciclo de vida del proyecto, no se identifican metodologías formales aplicadas o documentadas.
- El flujo de trabajo se adapta a la necesidad del proyecto y la experiencia del equipo.

5. Nivel de documentación.

- El nivel de documentación del equipo es señalado como una de las grandes falencias en el desarrollo de proyectos.
- Para los proyectos donde no se alcanza la formalización, no se genera documentación, lo cual dificulta la transferencia de conocimiento y, por lo tanto, continuidad de los proyectos.
- La deuda técnica más evidente recae sobre la falta de registro de procesos de negocio, arquitectura y procedimientos técnicos.

6. Impacto y limitaciones del Launchpad de Databricks.

- El Launchpad utilizado para la exploración de datos y acceso al Core Data Lake de la empresa impone ciertas limitaciones operativas, entre las cuales se destaca la imposibilidad de crear clústeres específicos para trabajo en producción.
 - Esto es generado de la falta de permisos y control sobre los recursos utilizados.
 - Es importante recalcar que esta imposibilidad genera costos innecesarios y limita la capacidad de aplicar CI/CD de forma efectiva.

7. Cumplimiento de métricas y estándares (*AI Readiness Criteria*).

- Respecto al AI Readiness Criteria, es necesario remarcar que en etapas tempranas de los proyectos no es tomado en cuenta. Esto es debido a la variabilidad y constancia de los requerimientos que se elevan en los proyectos.
 - La constancia y manejo de requerimientos crea una variabilidad constante en el código realizada por el ingeniero, lo que dificulta la integración del AI Readiness Criteria.
- El Launchpad de Databricks, a su vez, presenta limitaciones en las cuales se puede destacar: ambiente de producción, limitación de integración de librerías, no se encuentra verificado completamente por Infosec, entre otros.

8. Calidad del código y herramientas utilizadas.

- No existe un proceso automatizado para garantizar la calidad del código. Este mismo queda a criterio y apreciación del ingeniero.
- Ocasionalmente se utilizan herramientas como pueden ser Black y Isort.

9. Seguridad, confiabilidad y sostenibilidad. (Infosec)

- El equipo de AI Engineering no tiene acceso a esta información para conocer si los procedimientos actuales cumplen las normas de seguridad establecidas, este trabajo es directamente realizado por el equipo encargado del Launchpad.
 - El equipo de Ingeniería de Inteligencia Artificial únicamente consume datos de la plataforma del Launchpad.
- La desconexión existente entre Infosec y el equipo de Ingeniería de Inteligencia Artificial genera una barrera para incorporar y validar requisitos de seguridad desde etapas tempranas del desarrollo.
 - A su vez, la experimentación es minimizada al no conocer precisamente que componentes, librerías o procedimientos son correctos según los lineamientos de Infosec.

10. Aplicación de Code Review.

- No se considera como una práctica formal ni constante en el desarrollo de los proyectos de la entrevistada.
- Se prefiere aplicar solo al momento de fusionar el código en el *main Branch*, ya que durante el desarrollo hay constantes cambios que implican que esta práctica sea ineficiente.
- La entrevistada menciona que el proceso puede perder valor cuando la persona revisora no comprende el contexto del negocio, y que muchas veces el beneficio no justifica el esfuerzo invertido.
- Se considera que existen herramientas de inteligencia artificial que pueden facilitar de gran manera la aplicación de *code review*.

4.1.1.3. Análisis de hallazgos de las entrevistas

El siguiente análisis busca integrar los resultados obtenidos de las entrevistas de los *AI Engineers* en cuestión. El objetivo es evidenciar cuáles son las principales deficiencias, prácticas actuales y cumplimiento de estándares de calidad de la organización.

4.1.1.3.1. Deficiencias técnicas y operativas observadas.

Las entrevistas revelan que los ingenieros entrevistados presentan ciertas deficiencias técnicas y operativas en la gestión actual de los proyectos. Entre las mismas destaca la falta de planificación en fases exploratorias, la ausencia de aplicar un marco de trabajo estandarizado y la escasa documentación a lo largo de ciclo de vida de estos.

4.1.1.3.2. Procedimientos y metodologías actuales del desarrollo de proyectos E2E.

Según los entrevistados, el proceso de desarrollo de proyectos E2E en el equipo es principalmente informal, ambos describen que la planificación inicia con una necesidad planteada por el negocio. Posteriormente, el equipo inicia una fase de exploración de datos, análisis preliminar y cuando se llega a un punto de validación del alcance, se operacionaliza según *frameworks* internos de la organización.

En este flujo de trabajo se puede visualizar que existe una ausencia de metodologías ágiles en el proceso de planificación, y que de igual manera, el código desarrollado no sigue puntualmente un desarrollo que se encuentre bajo los estándares del *AI Readiness Criteria*.

Es importante recalcar que, al no aplicar métricas de calidad, una planificación ágil y no mantener procesos documentados, cuando los proyectos son escalados es posible transicionar un gran nivel de deuda técnica, lo cual impide el proceso de DevOps y, por lo tanto, es más complicada la integración con el equipo de SRE.

Es necesario destacar que existe una falta de claridad para la gestión de proyectos, ya que, en múltiples ocasiones, no se conocen concretamente las necesidades de la Unidad de Cuidado Familiar; esto debido a que no se conoce específicamente el trabajo que debe realizar un *AI Engineer*, lo cual es consecuencia de no presentar un marco de trabajo que refleje la generación de valor de estos profesionales.

4.1.1.3.3. Cuellos de botella en el desarrollo de proyectos

Los principales cuellos de botella se presentan en las dos etapas iniciales de los proyectos. Estos corresponden tanto a la planificación para la generación de requerimientos con el negocio como en la fase de exploración de datos.

Se identifica que, en la planificación, la falta de claridad por parte del cliente en cuanto a requerimientos dificulta definir con precisión los objetivos del proyecto, impidiendo la eficiencia del ciclo de vida de este.

La fase de exploración de datos se encuentra en un cuello de botella constante, debido al acceso a la información, la disponibilidad de los datos es significativa para generar los requerimientos del negocio. De igual forma, la documentación es escasa, por lo que cada requerimiento debe seguir un proceso de búsqueda con expertos por información que no se encuentra con facilidad, dificultando la claridad de los datos que deben utilizarse para la solución.

4.1.1.3.4. Nivel de documentación, automatización y deuda técnica

En ambas entrevistas, se puede llegar a la conclusión sobre la falta de documentación formal en las diferentes etapas del ciclo de vida de los proyectos. Este problema es considerado la mayor deuda técnica del equipo en el cual se debe indicar que es un problema en proyectos escalados, como en pruebas de concepto.

Los procedimientos manuales, *frameworks*, errores, fases exploratorias y los proyectos no contienen una documentación esperada, por lo que esto compromete la formalización de aquellos como la sostenibilidad de estos a largo plazo. A su vez, es necesario recalcar que esto limita la transición de conocimiento y de proyectos.

4.1.1.3.5. Limitaciones en la integración con el equipo de SRE

No se describen conflictos o limitaciones con el equipo de SRE, sin embargo, sí se identifican ciertas barreras de alineación técnica y estándares que deben ser incorporados en el equipo de Cuidado Familiar. SRE opera con su propio conjunto de requerimientos, lo cual implica que el equipo de Cuidado Familiar debe contar con ciertas especificaciones, a fin de que sus proyectos sean aptos para la integración con operaciones.

El ciclo DevOps no se cumple específicamente, ya que los proyectos llegan a ser implicados en producción una vez finalizada la prueba de concepto y, por ende, escalado. Esto quiere decir que no se integra continuamente mediante *Continuous Integration* y *Continuos Delivery*.

De igual forma, existe una dependencia constante al equipo Cuidado Familiar después de que se traslada el proyecto al soporte con SRE. Esto es debido a que el proceso de transición se realiza cuando este se encuentra en fases avanzadas, la documentación es escasa y se requiere de un proceso de *onboarding* amplio con los ingenieros para su integración.

4.1.1.3.6. Limitaciones con el gestor del Launchpad de Databricks.

Ambas entrevistas coinciden en que el Launchpad utilizado por el equipo presenta ciertas limitaciones técnicas significativas. Entre estas, se pueden destacar:

- La imposibilidad de crear *clústeres* personalizados que permitan al equipo desplegar soluciones con CI/CD automatizado. Esto genera costos mayores a la unidad de Cuidado Familiar, ya que se utiliza un clúster para todos los despliegues del equipo, el cual es más caro y menos personalizable en soluciones.
- No hay un ambiente de producción, lo cual aumenta la generación de errores y una nula diferenciación con el ambiente de desarrollo. Lo cual también imposibilita el despliegue continuo y la integración con la metodología DevOps.
- Se indica que el equipo no cumple ciertos requerimientos con *IRisk*, lo cual impide cumplir con los requerimientos de Infosec y con el equipo de SRE, ya que es necesario cumplir con estas pautas para que la aplicación sea denominada como confiable y segura.

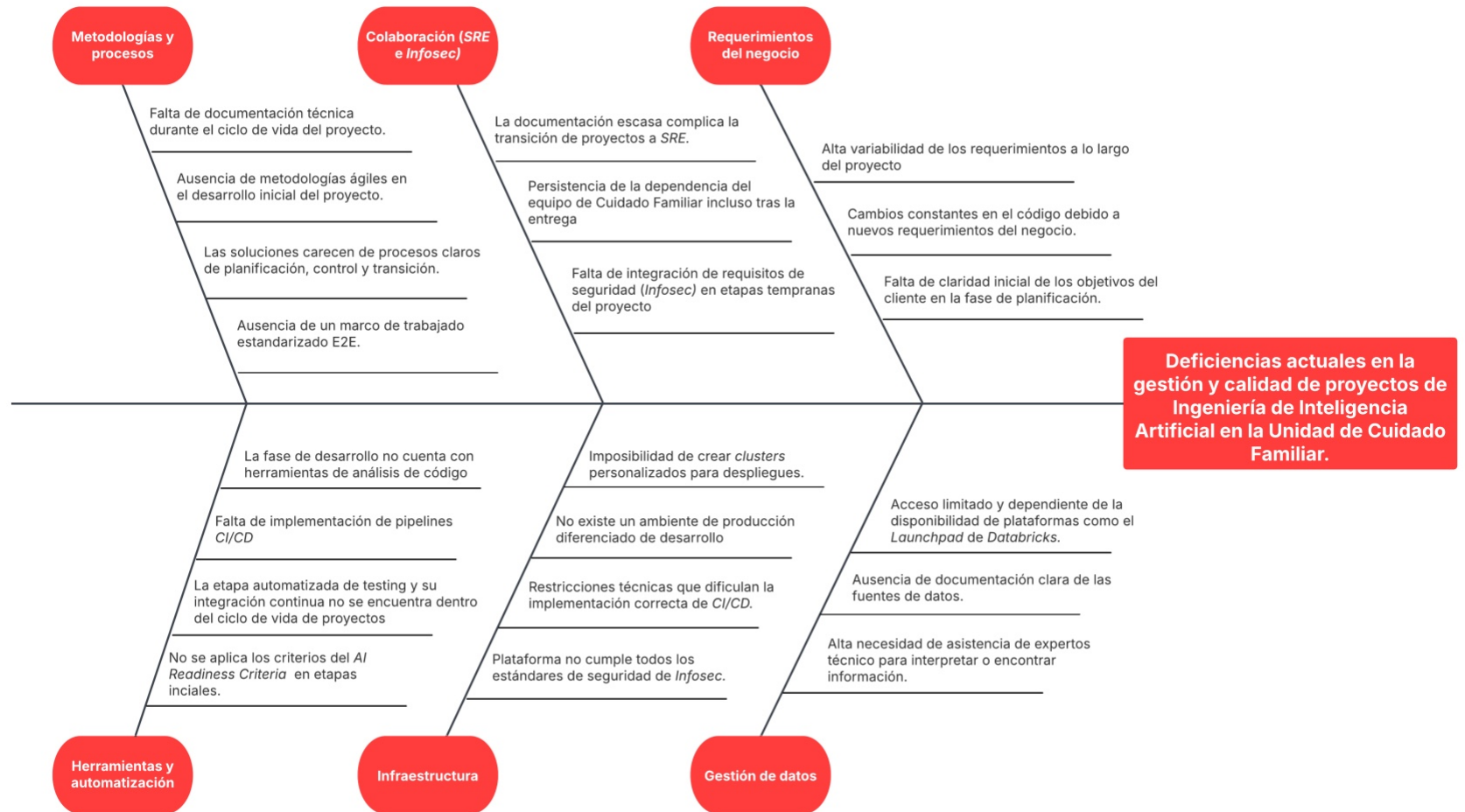
4.1.1.3.7. Nivel de cumplimiento con el AI Readiness Criteria

En las entrevistas se confirma que el AI Readiness Criteria no es integrado o aplicado formalmente en etapas iniciales del proyecto. De igual forma, estos criterios son aplicados cuando es formalizado y recibe presupuesto para su integración en operaciones.

La razón principal de que el AI Readiness Criteria no sea aplicado en la fase de desarrollo es porque existe una alta variabilidad de requerimientos por parte del negocio, lo cual provoca una constante modificación del código. Sin embargo, esto genera una deuda técnica constante que es preferible gestionar antes de entrar en etapas de producción.

4.1.2. Diagrama Ishikawa de la situación actual del equipo de Ingeniería de Inteligencia Artificial

Figura 10: Diagrama Ishikawa de causas raíz de la situación actual del equipo.



Fuente: elaboración propia (2025).

4.1.3. Revisión documental interna

4.1.3.1. Documentación de AI Readiness Criteria

El AI Readiness Criteria es un conjunto de lineamientos internos de la organización de Ingeniería de Inteligencia Artificial, que definen la preparación técnica, documental y operativa que deben seguir las soluciones informáticas de Inteligencia Artificial para su integración en producción. Por este motivo, para desarrollar un marco de trabajo estandarizado de DevOps CI/CD, es necesario seguir estas métricas y lineamientos según la naturaleza del proyecto. El AI Readiness Criteria permite alcanzar los estándares calidad, seguridad y sostenibilidad necesarios para las soluciones informáticas realizadas de la organización, por lo que actúa como un mecanismo de verificación tanto para auditorías como para el soporte en operaciones. En el siguiente apéndice se puede visualizar más información sobre esta documentación: Apéndice K. Revisión documental interna de las especificaciones del AI Readiness Criteria.

Dentro de la documentación revisada del AI Readiness Criteria, se identifican varios componentes que deben cumplirse para lograr la completitud de las soluciones de Inteligencia Artificial, los cuales se indican, a continuación:

Tabla 12: Componentes del AI Readiness Criteria

Categoría	Componente del AI Readiness Criteria
Documentación técnica	<ul style="list-style-type: none"> • Parámetros de código y marco de trabajo para el desarrollo de este. • Modelo de soporte del proyecto, incluyendo los roles que desempeñan los involucrados. • Código en repositorio en GitHub con control de versiones. • Documentación y configuración de CI/CD, incluyendo el desarrollo de <i>pipeline</i>, <i>branching</i> y <i>testing</i>. • Presencia de <i>test cases</i> y cobertura de pruebas (unitarias y de integración). • Documentación técnica almacenada en Confluence.
Cumplimiento normativo	<ul style="list-style-type: none"> • Validación de cumplimiento con políticas de Infosec de la empresa. (Presencia, en progreso o no implementado) • Uso de herramientas aprobadas tales como SonarQube, Snyk, GitHub, entre otras. • Administración de accesos a las herramientas, proyecto o aplicación. • Gestión formal de cambios documentadas en Confluence. • Gestión en Jira con enlaces a versiones y despliegues.

Categoría	Componente del AI Readiness Criteria
Procesos operativos	<ul style="list-style-type: none"> • Documentación de la transferencia al equipo de SRE con lista de soporte y monitoreo. • <i>Onboarding</i> para nuevos miembros, incluyendo accesos, ambientes, documentación y dependencias. • Configuración y documentación de credenciales y GitHub secrets. • Existencia y documentación de los ambientes del proyecto, tomando en cuenta desarrollo, pruebas y producción. • Documentación del proceso E2E del flujo funcional, operativo y de gestión de datos.
Métricas de calidad.	<ul style="list-style-type: none"> • Integración con SonarQube. • Cobertura mínima de pruebas automatizadas. • Historial de despliegues exitosos. • Métricas de negocio (impacto financiero, criticidad, entre otros).

Fuente: elaboración propia (2025).

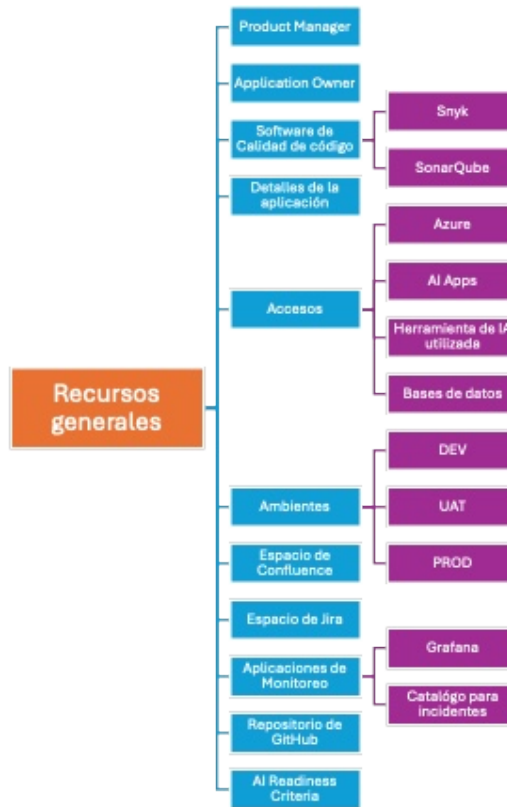
4.1.3.2. Documentación interna de la empresa de un proyecto actual en operacionalización del equipo de Ingeniería de Inteligencia Artificial

En el siguiente apartado, se evalúa un conjunto de artículos internos de la empresa, los cuales se encuentran en Confluence. Estos documentos hacen énfasis en cómo es el proceso de transición de un proyecto de Cuidado Familiar al equipo de SRE. Esto con el fin de verificar cuáles son los puntos claves tanto técnicos como operativos y administrativos para que el equipo de operaciones pueda dar soporte a un proyecto de forma efectiva. En el siguiente anexo se puede visualizar la información pertinente: [Apéndice I. Revisión documental interna de un proyecto en operación del equipo de Ingeniería de Inteligencia Artificial.](#)

Cabe recalcar que, por razones de confidencialidad, no se brinda el nombre de este proyecto, sin embargo, radica en un modelo de Inteligencia Artificial, el cual se consume mediante el API de OpenAI, que puede comprender y generar texto en lenguaje natural. Esta herramienta es considerada como un asistente virtual, el cual responde preguntas, redacta textos, resume información, analiza datos y distintas funciones de inteligencia artificial generativa.

La documentación del proyecto es un conjunto de diferentes secciones en las que es posible verificar cuáles son los principales ejes de la documentación, esto en relación con cuáles son más importantes para el equipo de SRE, con el fin de brindar soporte de forma efectiva. A continuación, se presenta el modelo de soporte del proyecto y la estructura general:

Figura 11: Estructura de modelo de soporte general de un proyecto interno en producción



Fuente: elaboración propia (2025).

4.1.3.2.1. Confiabilidad documentada de la aplicación

Esta documentación cuenta con un apartado de fiabilidad, donde se identifica cuáles son los SLAs de la aplicación, así como documentación sobre sus componentes. En este apartado, se destacan los siguientes puntos:

- Descripción de cómo deben funcionar las métricas de desarrollo de la aplicación, evidenciando los parámetros esenciales para que la aplicación funcione correctamente.
- Los procedimientos de seguridad deben estar aplicados para cualquier ambiente, ya sea en términos de *testing* o de desarrollo.

- La documentación abarca los procedimientos donde la aplicación falla, destacando atributos como:
 - Fecha
 - *Downtime* de la aplicación
 - Descripción
 - Estatus
 - Acciones realizadas
 - Puntos a tomar en cuenta

- Toda regla de disponibilidad, caídas y periodos estables se encuentran documentados.
- La documentación abarca accesos y disponibilidad de la aplicación en los siguientes ámbitos:
 - Contraseñas y accesos a usuarios
 - *Front-end*
 - *Back-end*
 - OpenAI
 - Generative AI
 - Bases de datos.
- Se describen los requerimientos del negocio, en los cuales destacan apartados tales como:
 - Expectativas del negocio
 - Motivación por la aplicación
 - Impacto métrico en el negocio
 - Impacto financiero y reducción de costos.
- Existen descripciones de cómo debe ser la confiabilidad de las fuentes de los datos, se debe verificar la data según los componentes que se vean integrados.

4.1.3.2.2. Arquitectura y diseño

Este proyecto cuenta con documentación sobre la arquitectura y diseño de la aplicación desarrollada. En este apartado, se destacan los siguientes puntos:

- Se cuenta con diagramas a Alto Nivel de la aplicación, así como diagramas técnicos.
- Los *test cases* se encuentran documentados
- Componentes de la aplicación como métodos, *endpoints*, *urls*, recursos y parámetros se encuentran debidamente documentados.
- Las funciones del código se encuentran documentadas.
- Existe un diagrama de SQL, el cual explica la estructura de este, ya sean las relaciones existentes o cómo está construido.

4.1.3.2.3. Documentación de administración de cambios

En esta sección de la documentación, se puede valorar una guía general del proceso para realizar cambios en el proyecto, ya sea en ambiente de producción o en desarrollo.

En cuanto a la documentación de la administración de cambios, se destaca:

- El manejo de despliegues en producción.
- La frecuencia de los despliegues y su implementación según diversos factores.
- Se describen las responsabilidades y roles para la administración de cambios, tomando en cuenta los siguientes roles:
 - SRE
 - Equipo de desarrollo de producto y de aplicaciones
 - Administrador de versiones
 - Administrador de *hotfix/rollback*.
- De igual forma, se describen los tipos de cambios que existen.
 - Normal
 - Emergencia
 - *Standard*.

4.1.3.2.4. Proceso de desarrollo

En esta sección, se valora el desarrollo del proyecto, este mismo tiene un enfoque en CI/CD, por lo que se valoran múltiples prácticas que buscan cumplir con las métricas del AI Readiness Criteria. A continuación, se destaca cuáles son los procedimientos principales del proceso de desarrollo:

1. El proyecto se construye mediante GitHub, la cual es la herramienta principal para el desarrollo del código.
2. Se implementa una estrategia de *branching* mediante un Git Flow definido.
3. El código debe contener un análisis con SonarQube que se encuentre automatizado dentro del *pipeline*.
4. Los *Unit test* deben ser automáticos y, a su vez, mandatorios.
5. El procedimiento de Code Review solo se realiza si el *pipeline* cumple con los procedimientos automatizados de análisis de código y *unit tests*.

4.1.3.2.5. Análisis estratégico de la documentación

Tabla 13: Análisis de la documentación interna de un proyecto en producción de Ingeniería de Inteligencia Artificial.

Sección de documentación	Práctica técnica u operativa	Aplicación para el desarrollo del marco
Confiabilidad y métricas	Documentación de SLA, fallos de la aplicación, tiempos de inactividad, acciones realizadas y análisis de disponibilidad.	Plantilla estándar con historial de errores, SLAs y guía de acciones ante fallos para cada proyecto.
Arquitectura técnica	Diagramas técnicos, funciones y estructura de SQL, <i>endpoints</i> y relaciones entre componentes.	Estándar mínimo de documentación de la arquitectura técnica de un proyecto, ya sea diagrama técnico de la aplicación, relación de componentes y documentación de bases de datos.
Administración de cambios	Definición de roles y responsabilidades, tipos de cambio y procedimientos formales.	Una sección formal para el control de cambios con roles definidos.
Ciclo de desarrollo DevOps CI/CD	Uso de GitHub con estrategia de GitFlow, pruebas unitarias automatizadas, prueba de calidad del código y <i>code review</i> en etapas de producción.	Estandarización de los procedimientos que debe de incluir el <i>pipeline</i> CI/CD a la hora de ejecución.
Vinculación con los objetivos del negocio	Registro del propósito del proyecto, impacto en métricas de negocio, expectativas y motivación financiera.	Se requiere un documento que describa el valor entregado mediante la aplicación, y cómo esta genera impacto en el negocio.

Fuente: elaboración propia (2025).

4.1.3.3. Documentación interna de un proyecto de Ingeniería de Inteligencia Artificial desarrollado para la Unidad de Cuidado Familiar

Esta sección estudia cómo un proyecto desarrollado por el equipo de Ingeniería de Inteligencia Artificial para Unidad de Cuidado Familiar es operacionalizado. Se desea visualizar el procedimiento E2E de cómo se integra un proyecto con el equipo de operaciones y cómo se desarrolla el trabajo de un *AI Engineer* en todo el flujo de trabajo. De igual manera, esta sección busca definir cualidades técnicas y operativas necesarias para el diseño del marco de trabajo y que estas mismas se encuentren validadas dentro de las métricas del AI Readiness Criteria. En el siguiente apéndice se puede visualizar esta documentación: Apéndice J. Revisión documental interna de un

proyecto en operación del equipo de Ingeniería de Inteligencia Artificial desarrollado para la Unidad de Cuidado Familiar.

4.1.3.3.1. Documentación de información y datos

Dentro de la documentación de este trabajo, es posible encontrar información sobre cómo se está construida la aplicación, específicamente sobre su modelo de base de datos utilizado. En este caso, se realiza una identificación de todas las fuentes, tablas, columnas y datos importantes dentro del desarrollo del proyecto, entregando documentación sobre el funcionamiento de los datos para este.

A su vez, existe una sección precisamente para el proceso de *data engineering*, en el cual se puede visualizar el proceso de ETL aplicado en el proyecto. Esta información va ligada específicamente a la idea del ingeniero de cómo aplicar un marco funcional en su proyecto, tomando en cuenta los pasos para el desarrollo de un proceso de *data engineering* efectivo. Para este caso, se encuentra la documentación de la transición del código de *Notebooks* no oficiales a Databricks Asset Bundles.

En el apéndice P, se encuentra más a detalle la construcción de la aplicación del proyecto que se estudia en esta sección de documentación interna.

4.1.3.3.2. Documentación de prueba de concepto realizada

Previamente, en el análisis de las entrevistas sobre la situación actual del equipo, uno de los puntos destacados de la deuda técnica es precisamente la ineficiencia de la documentación en los proyectos del equipo, sobre todo, la inexistencia de documentación en aquellos que se consideran pruebas de concepto.

Dentro de los puntos revisados, se encuentra una sección de cómo se ha desarrollado la prueba de concepto. Esta documentación incluye secciones que valorizan el proyecto, por lo que su transición en producción es más rentable y, por ende, más realizable. Este apartado de la documentación incluye cómo se encuentra desarrollado el proyecto, la infraestructura de este, la gestión de sus datos, herramientas y valor para el negocio.

La documentación de las pruebas de concepto es primordial para mostrar al negocio el valor que un proyecto de automatización, ingeniería de datos o de inteligencia artificial puede brindar.

Se documentan las funciones realizadas dentro del proyecto, modelos desarrollados, fórmulas e información sobre el proceso de manufactura. Cabe recalcar que el proceso de manufactura es explicado por el negocio, sin embargo, cada ingeniero debe conocer el funcionamiento del proceso de manufactura, componentes, líneas de trabajo, entre otros, por lo que su documentación es esencial para el entendimiento de los proyectos.

4.1.3.3.3. Infraestructura y accesos

Esta sección de la documentación explica de forma visual y precisa las diferentes herramientas que componen la aplicación de este trabajo. Esta visualización se basa en un marco de referencia, el cual explica cómo la información, herramientas y componentes se relacionan para llevar a cabo el proyecto. De esta forma, se visualiza el ciclo de vida implementado hasta la entrega de valor.

A continuación, se especifican las secciones pertenecientes a la documentación de infraestructura y accesos:

- Grupos de Azure y consumo de información.
- Documentación sobre el Launchpad de Databricks.
- Repositorios de GitHub.
- Gestión de la calidad de código mediante SonarQube.
- Gestión de requerimientos y *sprints* mediante Jira.
- Gestión de diseño de *UI/UX* mediante Figma.
- Integración con herramientas de *Data Science* y aplicación de Inteligencia Artificial.

4.1.3.3.4. Estrategia de CI/CD

Esta sección describe la estrategia de CI/CD que busca integrar el equipo de este proyecto para la escritura, fusión y despliegue del código de la aplicación en desarrollo y producción. Este cuenta con dos componentes principales: un *pipeline* en Azure Machine Learning y una aplicación de Inteligencia Artificial, los cuales se encuentran en distintos repositorios en GitHub.

La estrategia que se desarrolla en ambos repositorios busca seguir un Git Flow para todos los *branches* en uso, los cuales dividen el trabajo según el propósito. A continuación, se explican los tipos de *branches* que se implementan en este proyecto:

Tabla 14: Branching Strategy

Branch	Pautas
<i>Main</i>	<ul style="list-style-type: none"> • Representa el estado en producción del código • Todo lo que se encuentre en este <i>Branch</i> fue validado y está listo para ser utilizado por los usuarios finales. • Solo se actualiza con código estable, tras un proceso de pruebas y validaciones previas.
<i>Develop</i>	<ul style="list-style-type: none"> • Se crea a partir del <i>main Branch</i>. • Representa un entorno de desarrollo o “pre-producción”, donde se integran funcionalidades que aún no están listas para producción o no han sido validadas. • Una vez que las funcionalidades son validadas y se consideran estables, este <i>Branch</i> se fusiona con <i>main</i>.
<i>Feature</i>	<ul style="list-style-type: none"> • Se crea a partir del <i>develop branch</i>. • Se usa para desarrollar nuevas funcionalidades. • El nombre del <i>Branch</i> debe reflejar una historia de usuario. Por ejemplo: feature-filtro-búsqueda. • Una vez completada la funcionalidad, se fusiona con <i>develop</i>.
<i>Hot-fix</i>	<ul style="list-style-type: none"> • Se crea a partir del <i>main branch</i> para corregir errores críticos en producción. • Se debe nombrar usando una historia de usuario relacionada al problema. • Una vez corregido, se valida y se fusiona con <i>main</i> y <i>develop</i>, con el fin de sincronizar ambos <i>branches</i>.

Fuente: elaboración propia (2025).

4.1.3.3.5. Documentación administrativa

Este apartado de la documentación contiene recursos del uso administrativo de la aplicación, donde se encuentran los roles específicos del proyecto, los costos incurridos y el valor proporcionado. La documentación incluye un plan de *onboarding* para nuevos miembros del equipo, siendo fundamental para la transición de proyectos o, en su defecto, efectividad en el proceso de aprendizaje de un nuevo miembro en este.

La documentación administrativa, por lo general, se encuentra gestionada por el *product owner* del proyecto, por lo que en esta sección se encuentran los *user stories* para el desarrollo de este.

4.1.4. Análisis comparativo entre las entrevistas realizadas y la documentación interna

Con el fin de obtener una visión estructurada del estado actual del equipo de Ingeniería de Inteligencia Artificial y su relación con los procesos de DevOps CI/CD, AI Readiness Criteria y su gestión de proyectos de índole informática y de inteligencia artificial, se realiza un análisis comparativo entre la información recopilada de las entrevistas a miembros del equipo y la revisión documental interna del equipo. Este análisis permite identificar las falencias del procedimiento actual, lo cual impulsa el diseño de un marco de trabajo estandarizado.

Tabla 15: Análisis comparativo entre entrevistas y la documentación interna

Aspecto evaluado	Análisis de la entrevista	Resultado de la documentación interna
<p>Metodologías para el desarrollo E2E de los proyectos.</p>	<p>El proceso es unificado, la mayoría de los proyectos reciben requerimientos y conforme las necesidades del negocio se desarrollan para una solución informática. Cada integrante trabaja con sus propios criterios, tecnologías y procedimientos.</p> <p>La diferencia de soluciones provoca el entendimiento erróneo de las soluciones que el equipo es capaz de brindar a la Unidad de Cuidado Familiar.</p> <p>Es necesario recalcar que no se presenta una documentación estándar que debe ser integrada en el desarrollo de cualquier tipo de proyecto por el equipo. Las entrevistas evidencian una carencia clara de la estandarización documental, por lo que se reconoce un impedimento para la correcta escalabilidad de proyectos.</p>	<p>La documentación interna no presenta un proceso E2E para la aplicación de proyectos de <i>software</i>, sin embargo, si presenta ciertos lineamientos que deben seguirse, los cuales abarcan desde el cumplimiento normativo, documentación técnica, monitoreo, integración con el equipo de operaciones, entre otros.</p> <p>La documentación busca estandarizar y registrar los parámetros de código utilizados, el Modelo de Soporte de cada proyecto, la configuración de la aplicación y su arquitectura, el CI/CD implementado, el modelo de datos utilizados, así como la documentación técnica correspondiente para entender la solución informática brindada.</p>

Aspecto evaluado	Análisis de la entrevista	Resultado de la documentación interna
Nivel de documentación	<p>No existe una documentación formal, por lo que cada miembro del equipo es encargado de registrar requerimientos, cambios y decisiones técnicas de los proyectos.</p> <p>Es correcto señalar que no se brinda una plantilla de cómo deben documentarse datos, patrones de código, arquitectura u otros elementos que guíen al usuario para conocer cómo se ha desarrollado dicho proyecto.</p>	<p>La documentación interna analizada visualiza que sí se cumplen con esfuerzos formales para el desarrollo de documentación administrativa y técnica. Esto se evidencia mediante el uso de plantillas técnicas, documentación de arquitectura, documentación del modelo de datos, administración de cambios, aplicación de CI/CD y métricas de éxito. Cada documentación consta de diferentes variaciones según el tipo de proyecto, ya sea de integración con inteligencia artificial, ingeniería de datos o ciencia de datos, sin embargo, existen los lineamientos que son necesarios para produccionalizar un proyecto. De igual forma, la documentación, al variar según la estructura, detalle o propósito, impide establecer un estándar unificado que sea reutilizable para el equipo. Por esta razón, es recomendable establecer cuáles son esos requerimientos mínimos a desarrollar.</p>

Aspecto evaluado	Análisis de la entrevista	Resultado de la documentación interna
Implementación de CI/CD	<p>Las entrevistas realizadas coinciden en la ausencia de una implementación formal del ciclo de CI/CD en los proyectos.</p> <p>Si existen pruebas por lo general se ejecutan manualmente. A su vez, no se aplica la ejecución de un <i>pipeline</i> automatizado en la mayoría de los casos.</p>	<p>Se documenta el uso de herramientas como GitHub integrado con un Gitflow, utilizando un <i>pipeline</i> automatizado para la gestión de pruebas unitarias, análisis de la calidad de código mediante SonarQube y el uso de Git Branch.</p> <p>La documentación señala una intención de aplicar el funcionamiento de un <i>pipeline</i> de CI/CD, sin embargo, aún no es un estándar para las soluciones de los <i>AI Engineers</i>.</p>
Integración con el equipo de SRE	<p>Se presentan barreras técnicas con SRE, debido a la poca documentación existente, no es posible generar un procedimiento efectivo a la hora de produccionalizar un proyecto.</p> <p>No existe un procedimiento formal de transición, por lo que, al no contar con los lineamientos necesarios para una transferencia a operaciones, existe una alta dependencia a los ingenieros de <i>AI Engineering</i> encargados del procedimiento ejecutado para la Unidad de Cuidado Familiar.</p>	<p>Es necesario elaborar un modelo de soporte, la definición de los SLAs, sesiones de <i>onboarding</i> técnico y documentación operativa.</p> <p>Con la definición y documentación de estos lineamientos, el equipo de operaciones empieza el trabajo operativo en los proyectos de Ingeniería de Inteligencia Artificial, donde es necesaria la documentación de todos los procedimientos, herramientas y procesos operativos que se realizan.</p>

<p>Interacción con el Launchpad de Databricks.</p>	<p>Las entrevistas reportan limitaciones técnicas del Launchpad de Databricks actual. Estas limitaciones son las siguientes: creación de despliegues en producción automatizados según el clúster utilizado, barreras para la aplicación correcta de CI/CD, poca claridad sobre lineamientos de seguridad y roles para gestionar las características técnicas del Launchpad.</p>	<p>La revisión documental confirma las carencias actuales del equipo, ya que mediante el AI Readiness Criteria es posible concluir la necesidad de ambientes de desarrollo y producción, documentación técnica, verificación con Infosec, integraciones con modelos de AI mediante el AI Factory o, en general, un uso más adecuado de las soluciones que buscan generar valor en producción.</p>
--	--	---

Fuente: elaboración propia. (2025).

4.1.5. Observación directa

Durante la fase uno, se realizó la observación directa de procedimientos de integración con miembros del equipo de Site Reliability Engineering y de las métricas del AI Readiness Criteria. Esta observación permite identificar elementos esenciales relacionados a la transición de proyectos hacia el equipo de operación, así como la ejecución de métricas y estándares aplicados.

4.1.5.1. Procedimiento de integración con el equipo de Site Reliability Engineering

A continuación, se describen los puntos clave identificados para la transición de un proyecto a operaciones:

1. Definición del modelo de soporte: consiste en que el equipo de AIE y el Product Owner del proyecto se reúne con el equipo de SRE para completar un documento base de soporte, el cual permite formalizar los acuerdos de operativos logrables por el equipo. Este documento tiene como objetivo:
 - Delimitar cuáles componentes del proyecto se encuentran dentro o fuera del alcance del equipo de operaciones.
 - Establecimiento de SLAs.
 - Definir los tipos de incidentes esperados, criticidad y tiempo máximo de resolución.
 - Identificar componentes críticos y puntos de fallo esperados.

Una vez que se define el modelo de soporte, se procede a realizar sesiones de *onboarding técnico* con el equipo de operaciones, para garantizar una comprensión general del proyecto. Sin embargo, hay que destacar que el equipo de operaciones no requiere del conocimiento en detalle del código fuente, a pesar de esto, sigue siendo el ingeniero de Inteligencia Artificial el encargado de *bugs* o del desarrollo de nuevas funcionalidades.

4.1.5.2. Observación directa de la aplicación de las métricas del AI Readiness Criteria

Como parte del proceso de observación directa realizado con el procedimiento de integración de un proyecto con el equipo de SRE, se evalúa la aplicación real de las métricas establecidas por AI Readiness Criteria. Para dicha integración, es necesario valorar cuáles son aspectos indispensables y cuáles aspectos se incluyen según el marco u objetivo que un proyecto implemente.

Durante la interacción con miembros del equipo, se evidencia que, si bien existe un entendimiento general sobre la necesidad de cumplir con este conjunto de criterios, su aplicación no se encuentra integrada de forma sistemática en todos los proyectos, por lo que no se desarrolla el estándar como tal.

Entre los hallazgos más relevantes, se destacan los siguientes:

1. Inicio con la integración con Infosec: la aplicación observada contaba con un *IRisk* en progreso, lo cual indica una intención de cumplir con los lineamientos de seguridad de Infosec, por ende, de la Organización. Por lo que este es considerado como uno de los requisitos para lograr la operacionalización con el equipo de SRE.
2. Se indica una documentación mínima de componentes importantes tanto como la arquitectura técnica, flujo de datos y herramientas utilizadas (Databricks, GitHub, Jira y Confluence).
3. Cobertura de pruebas y calidad de código: existen prácticas de *testing* y revisión de código, sin embargo, se concluye que estas herramientas no se encuentran automatizadas como parte del *pipeline* de la aplicación. Este procedimiento de automatización es uno de los principales en la evaluación y políticas en torno a las políticas CI/CD evaluadas en el AI Readiness Criteria.

Esta observación directa evidencia que las métricas del AI Readiness Criteria son conocidas, sin embargo, no se aplica de una forma estandarizada mediante CI/CD.

4.1.6. Revisión documental de marcos de referencia

4.1.6.1. Marco de referencia DevOps y Lean

La implementación de un marco de trabajo, el cual combine DevOps, siempre se encuentra en sinergia con los principios de *Lean*. Esta combinación entre prácticas busca mejorar tanto el rendimiento técnico como organizacional de los equipos de TI. Además, esta sinergia entre prácticas de *Continuous Delivery* y eficiencia operacional se respalda del estudio en el libro *Accelerate*, el cual identifica diferentes factores que distinguen a las organizaciones de un alto desempeño. En el siguiente apéndice se hace referencia a esta documentación: [Apéndice L. Revisión documental: The DevOps Handbook 2021.](#)

1. DevOps como medidor de desempeño

DevOps no se trata únicamente de la búsqueda de automatización y el uso de las mejores herramientas, sino que DevOps busca fomentar la cultura organizacional y la colaboración entre el desarrollo y operaciones, eliminando silos y facilitando ciclos

de retroalimentación más eficientes. Según Forsgren et al. (2018), el desempeño de los equipos de *software* es medible a través de cuatro métricas clave:

- Frecuencia de despliegue: corresponde a qué tan a menudo se despliegan cambios a producción.
- Tiempo de entrega de cambios: cuánto tiempo pasa un cambio desde el *commit* hasta el *main Branch*.
- Tiempo medio de recuperación: cuánto tiempo tarda el equipo en restaurar el servicio tras una falla.
- Tasa de fallos por cambios: el porcentaje de despliegues que causan errores.

Se considera que estas métricas permiten evaluar el impacto de las prácticas DevOps y son predictores sólidos del desempeño del equipo. Para la evaluación de las métricas, hay que recalcar que no existe un tiempo validado o estandarizado en el que se encuentre la eficiencia operativa según DevOps. A pesar de esto, existen ciertos métodos en los cuales es posible medir la efectividad para el equipo de Ingeniería de Inteligencia Artificial:

1. La frecuencia de despliegue se valora con el número de integraciones en el *main Branch*, donde se valora la frecuencia de los cambios integrados en el *pipeline* automatizado.
2. El tiempo de entrega de cambios se puede evaluar según la consistencia en que los cambios son revisados mediante *code review*, y la aplicación de tareas dentro del *backlog* en el que se pueda considerar integrar el *code review* dentro del tiempo de desarrollo de funcionalidades.
3. El tiempo medio de recuperación se debe evaluar según cada ejecución del *pipeline*, de acuerdo con su configuración y constancia de despliegue. Por lo general, su ejecución se encuentra configurada según los datos ingresen y la necesidad del negocio a su disposición. Esta métrica debe valorar mediante la visualización del *pipeline* los errores de integración en todo el ciclo de compilación, pruebas y despliegue, configurando una serie de etapas en las cuales será más sencillo verificar dónde se encuentra el error en el proyecto.
4. Para valorar la tasa de fallos por cambios, es necesario considerar la forma en que el *pipeline* se encuentra construido a partir de cuáles funcionalidades son claves para que el proyecto brinde los resultados esperados. Para realizar esta evaluación, es necesario contar con un procedimiento de desarrollo sólido, el cual se vea reflejado en GitHub y brinde los resultados que proporciona cada despliegue, por ende, los errores que se encuentran en el mismo.

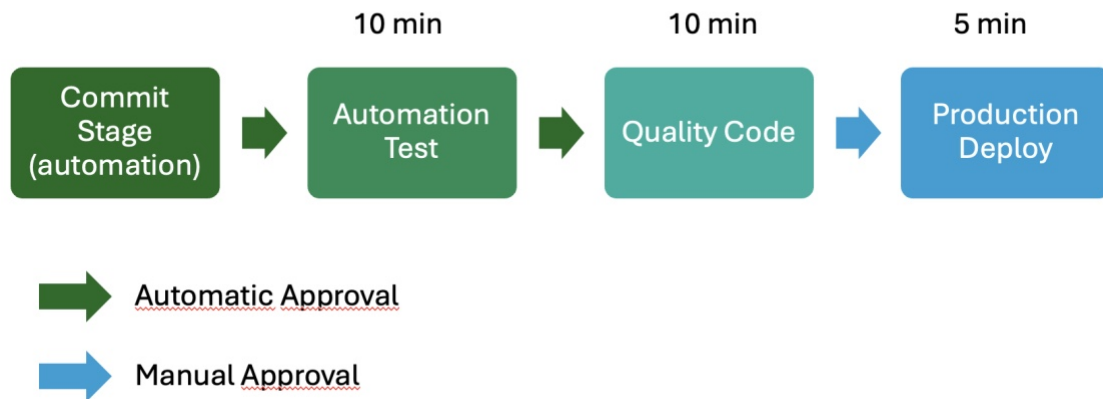
2. Lean Management

Lean se fundamenta, principalmente, en la manufactura, aportando en principios que permiten eliminar desperdicios, mejorar el flujo de trabajo y generar aprendizaje continuo. Este contexto es aplicable a áreas de TI, específicamente para los proyectos de Inteligencia Artificial, se pueden aplicar prácticas tales como las siguientes:

1. **Visualización del trabajo en curso:** la adopción de herramientas, como tableros Kanban en Jira y su integración con procesos Scrum, permite a los miembros del equipo identificar bloqueos y prioridades en tiempo real.
2. **Límites al trabajo en curso (*WIP* o *Work In Progress*);** establecer los límites de WIP es fundamental para la priorización de los proyectos de *software* y, por ende, de Inteligencia Artificial. Según su implementación, se establece un límite explícito al número de tareas abiertas por ingeniero, por ejemplo: en progreso, *code review* y fase de *testing*. Aplicar la limitación de WIP permite una integración a las metodologías ágiles y seguir un camino estructurado en el desarrollo de proyectos de esta índole.
3. **Gestión del flujo mediante ciclos de retroalimentación rápidos:** se considera que el flujo mediante ciclos de retroalimentación rápidos beneficia el proceso de desarrollo de un proyecto de *software*. En el equipo de Ingeniería de Inteligencia Artificial, la aplicabilidad de esta gestión podría observarse de la siguiente forma:
 - *Code Review* estructurado antes del *merge* al *main Branch*.
 - Despliegues frecuentes en ambientes controlados como *development* o *UAT*.
4. **Gestión del flujo de valor y el *Lead Time*:** la gestión del flujo de valor busca reforzar los principios Lean en entornos tecnológicos, enfocándose en la reducción del *Lead Time* y su integración como cultura organizacional. Este proceso enfatiza que el tiempo de entrega entre fases del ciclo de vida del proyecto debe ser reducido al máximo, las prácticas automatizadas en un *pipeline* evidencian la eficiencia del flujo de valor.

A continuación, en la siguiente imagen se ejemplifica el procedimiento de la gestión del flujo de valor según su valor en *Lead Time*, el cual debe ser integrado en proyectos de *software*, siendo un aspecto crítico el tiempo en el que se realizan las fases de la automatización del proyecto.

Figura 12: Gestión del flujo de valor mediante Lead Time



Fuente: elaboración propia. Basada en “A Technology Value Stream with a Lead Time of Minutes” de The DevOps Handbook (2021).

4.1.6.2. Marco de referencia de aplicación de CI/CD

La información pertinente a este marco de referencia se encuentra en el siguiente apéndice: Apéndice M. Revisión documental: Continuous Integration and Continuous Delivery (2023)

4.1.6.2.1. Introducción a CI/CD y su impacto en el ciclo de vida de proyectos de software

Continuous Integration y *Continuous Delivery* representan prácticas fundamentales en empresas en busca de la modernización o transformación digital; estos conceptos permiten la entrega frecuente, confiable y automatizada desde el entorno de desarrollo hasta el de producción. A continuación, se analiza un punto de partida para el entendimiento y la aplicabilidad de los conceptos de CI/CD enfocado en la mejora continua y la entrega de valor.

CI/CD es una respuesta técnica a los retos del desarrollo tradicional que se pueden visualizar en el equipo de Ingeniería de Inteligencia Artificial. En este sentido, CI se enfoca en la automatización y validación de la integración de código, mientras que CD se enfoca en la automatización y validación del código antes de su despliegue. Según el autor de *Continuous Integration and Continuous Delivery*, “La entrega continua no es un producto que pueda instalarse, sino una práctica que requiere organización, mantenimiento y adaptación constante” (Van Merode, 2023, p.4).

Esta perspectiva resulta especialmente relevante al analizar la situación actual del equipo de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar, ya que no se encuentra una idea clara de la implementación de CI/CD. En este caso, el provisionar de *pipelines AI Provisioner* del *AI Factory*, que genera *pipelines* para producción, *testing* y desarrollo. Sin embargo, varios de sus pasos y estructura no son utilizados o considerados por el equipo, lo que indica una falta de contextualización y personalización de *pipelines* para su realidad operativa. Además, *AI Factory* se encuentra más enfocado en el desarrollo de modelos de *Data Science*, mientras que un gran porcentaje de los proyectos en la Unidad de Cuidado Familiar son enfocados en *Data Engineering*.

Seguidamente, se desarrollan puntos de importancia para valorar el impacto del ciclo de vida de proyectos de *software*:

1. Vulnerabilidades de CI/CD

Continuous Integration and Continuous Delivery destaca los riesgos técnicos que se pueden sufrir de las vulnerabilidades de una mala aplicación de CI/CD. Estas limitaciones técnicas corresponden en problemas de seguridad, dificultad de rastrear fallos y falta de confianza en el desarrollo del *pipeline*. Según *Continuous Integration and Continuous Delivery*, por ejemplo, al integrar librerías de terceros, se tiene una exposición a problemas que resultan en fallos de seguridad y, por ende, un problema en la calidad de la solución desarrollada.

2. Organización y mantenimiento de pipelines

En el equipo de Ingeniería de Inteligencia Artificial, es común el desarrollo constante de distintas soluciones informáticas debido a los múltiples ámbitos y proyectos que se desarrollan. Por esta razón la adopción de prácticas de CI/CD, como de marcos de mejora continua, deberían variar en consecuencia. Según el libro de *Continuous Integration and Continuous Delivery*: “Después de crear un *pipeline*, se crea el segundo, se crea el tercer, y docenas de *pipelines que intentar cumplir con múltiples* pasos dentro del proceso de *CI/CD*” (Van Merode, 2023, p.5). Esta situación genera caos operativo y eventualmente es necesaria la reestructuración completa.

Según el libro de *Continuous Integration and Continuous Delivery*, es necesario considerar diferentes factores al organizar un *pipeline*, los cuales son descritos, a continuación:

1. **La arquitectura de la aplicación:** ya sea que la arquitectura sea monolítica o basada en microservicios. Esto debido a los múltiples tipos de requerimientos que una aplicación puede abarcar, en procedimientos del ciclo, tal como *testing* y despliegue, así como en general el

desarrollo de proyectos mediante herramientas del AI Factory, proyecto de *Data Engineering*, *Data Science*, entre otros que requiera el negocio.

2. **El flujo de trabajo del equipo:** es necesario contar con la estructura en la que está planteado el desarrollo de proyectos, en qué ámbitos trabaja el equipo y cuál es la frecuencia de los requerimientos del negocio. Además, es clave contar con una estrategia de *branches*, así como *test cases*, organización de despliegues y de *releases*. Estas decisiones deben ser documentadas e integradas en el diseño del *pipeline*.
3. **El número de pasos manuales:** es necesario analizar cuántos pasos o procedimientos manuales se realizan para la ejecución del *pipeline*, los cuales deben estar focalizados a la automatización y, si no es posible realizarlo en completitud, es necesario justificar el diseño técnico y cómo esto afecta el proceso del *pipeline*.
4. **La estructura del código:** es necesario conocer cómo el código se encuentra estructurado, este mismo puede consistir en analizar las siguientes variables:
 - a. Conocer donde se encuentra alojado el código, ya sea en *notebooks* o en código en repositorios.
 - b. Existencia de múltiples repositorios.
 - c. La organización del código en la aplicación.
 - d. Existencia de patrones de código o marcos de diseño.

Estos cuatro elementos son fundamentales de considerar a la hora de la estructuración de *pipelines*, lo cual es crítico debido a la amplia gama de proyectos en el equipo y las múltiples soluciones que pueden reflejarse, con flujos que varían significativamente en duración, tipo de datos, herramientas y criterios de validación.

4.1.6.2.2. Posicionamiento de CI/CD dentro del flujo de desarrollo

El concepto de *Continuous Integration* como de *Continuous Deployment* no corresponde a prácticas asiladas, sino que son componentes y fases dentro del ciclo de vida de los proyectos de desarrollo de *software*. El flujo de valor de este marco no reside únicamente en la automatización de tareas, sino que se ve ejemplificado en el tiempo de desarrollo y la entrega de este para mejorar la calidad del producto y dar visibilidad a la mejora continua de la solución.

La realidad de las empresas es que la mayoría de los equipos inician sin un marco estructurado para el desarrollo del valor en sus soluciones. Hablando específicamente de *software*, CI/CD actúa como el enlace entre el desarrollo, pruebas y entrega, estableciendo este flujo de valor constante. Este enfoque es especialmente importante para el desarrollo de proyectos de Inteligencia Artificial,

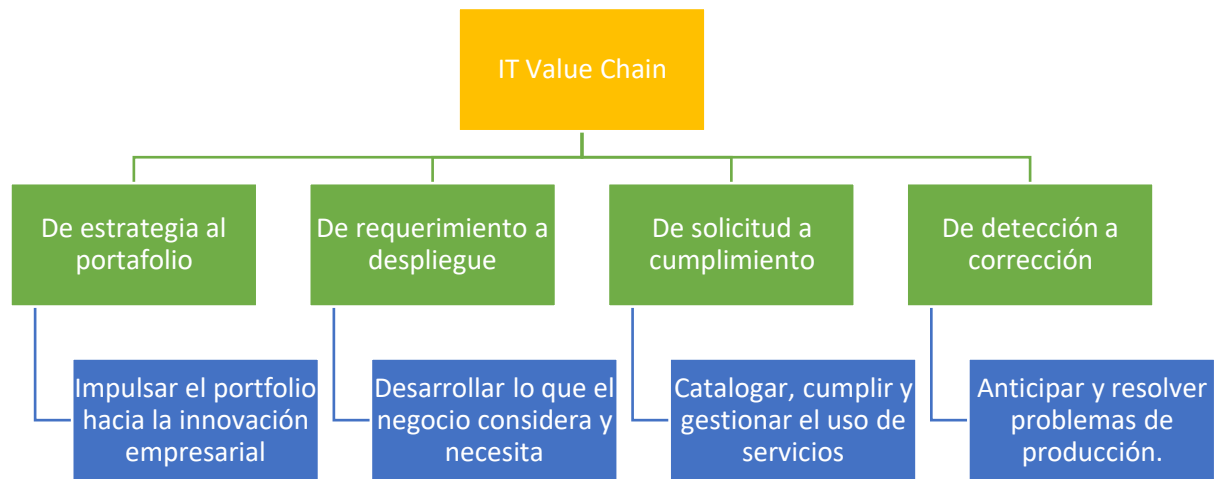
donde es necesaria la planificación de múltiples pasos para considerar una entrega lista para su producción.

El libro *Continuous Integration and Continuous Delivery* resalta que el marco de CI/CD debe considerar una práctica o implementación desde las fases más tempranas del desarrollo, viéndose reflejado en: “CI/CD debería ser integrado desde la fase inicial de planeación del proyecto, facilitando el flujo coherente entre los requisitos del negocio y los entregables técnicos. Esto asegura que la automatización y la validación acompañen en cada paso del desarrollo” (Van Merode, 2023, p.23). Esto implica que la construcción del *pipeline*, como el marco de trabajo propuesto en el proyecto, debe incluir pautas de CI/CD para ser correctamente aplicado.

CI/CD debe considerar desde un principio cuáles artefactos serán generados, cómo serán validados y en qué condiciones deben cumplirse. En el caso del equipo de Ingeniería de Inteligencia Artificial, los proyectos cuentan con cambios frecuentes de código, datos y modelos, permitiendo a CI/CD capturar los requisitos específicos de calidad y seguridad del diseño inicial.

El flujo de valor de CI/CD se encuentra estrechamente alineado con el *IT Value Chain*, el cual describe las funciones clave que debe cumplir un área de TI para generar la entrega de valor sostenible al negocio. Según el libro *Continuous Integration and Continuous Delivery*: “La cadena de valor de TI debe incluir prácticas que aseguren una conexión fluida, entre la planificación estratégica y la entrega técnica continua” (Van Merode, 2023, p.26). Esto quiere decir que CI/CD permite ejecutar de forma ágil los compromisos adquiridos con el negocio, mientras que, a su vez, habilita la retroalimentación rápida y una integración más fluida entre desarrollo, operaciones y estrategia. Para ejemplificar este flujo de valor, se presenta la siguiente figura:

Figura 13: IT Value Chain



Fuente: elaboración propia. Basada en “Positioning of CI/CD” de *Continuous Integration and Continuous Delivery* (2023).

4.1.6.2.3. *Application Lifecycle Management (ALM)*

CI/CD no debe ser visto como una secuencia de pasos para generar el flujo de valor, donde se aplique el aprendizaje continuo y la entrega. Según el libro *Continuous Integration and Continuous Delivery*: “La gestión del ciclo de vida de la aplicación requiere considerar todos los pasos de software, desde la entrega hasta el mantenimiento, integrando capacidades como el control de versiones, *testing*, monitoreo y despliegue” (Van Merode, 2023, p.26).

Es importante reconocer que el ALM puede relacionarse con la integración con DevOps, abarcando todas las fases de un proyecto. A continuación, se menciona cómo el CI/CD se integra con el desarrollo DevOps y, a su vez, con los proyectos de la Unidad de Cuidado Familiar:

Figura 14: Fases ALM

Fases ALM	Descripción
Planificación	Permite visualizar cuáles artefactos, componentes y pruebas deben aplicarse dentro del <i>pipeline</i> y cómo se manejarán las versiones del proyecto.
Desarrollo	Facilita la integración continua del código y habilitando entregas funcionales. Dentro del <i>pipeline</i> esto puede ser aplicable.
Despliegue	Garantizar despliegues repetibles, auditables, lo que fortalece el cumplimiento.
Mantenimiento	Permite automatizar procesos de corrección, monitoreo y mejora continua, alineando la evolución técnica con las necesidades del negocio.

Fuente: elaboración propia (2025).

El enfoque ALM es especialmente relevante para proyectos del equipo de Ingeniería de Inteligencia Artificial, donde los productos no son aplicaciones tradicionales, sino *pipelines* complejos que integran datos, modelos, validaciones estadísticas y componentes de seguridad. La aplicación de CI/CD permite asegurar que estos proyectos cuenten con una estructura controlada, que permita soluciones escalables y que cumplan con patrones de calidad.

CI/CD, a su vez, facilita y habilita la gestión basada en ciclos de *feedback* continuo, la incorporación de herramientas como GitHub Actions, Azure DevOps o Databricks Asset Bundles permiten agilizar la incorporación de las métricas de desempeño como AI Readiness Criteria, así como ciclos de desarrollo más cortos mediante la identificación de errores recurrentes, cobertura de *testing* y automatización.

Según *Continuous Integration and Continuous Delivery*: “La automatización no reemplaza la gestión del ciclo de vida, sino que la habilita, proporcionándole una columna vertebral técnica que respalda cada decisión y cada entrega” (Van Merode, 2023, p.26). Por tanto, una propuesta de marco estandarizado para proyecto de Inteligencia Artificial debe considerar CI/CD para una correcta aplicación de estandarización, no como únicamente mejoras técnicas aisladas, sino como un componente central en el ciclo de vida de proyectos que son desarrollados por el equipo.

4.1.6.3. Herramientas aplicables al ciclo DevOps CI/CD

4.1.6.3.1. Databricks Asset Bundles

Databricks Asset Bundles (DAB) representa una herramienta para la gestión y despliegue de proyectos en Databricks, este mismo es considerado una evolución en los equipos que desarrollan proyectos de ingeniería de *software*, ingeniería de datos o de inteligencia artificial. DAB busca incorporar las mejores prácticas de *software* mediante Databricks en una única estructura llamada “*bundle*”.

DAB permite al usuario experimentar con los datos mediante la conexión con un clúster y una sesión de Databricks. Su funcionalidad es primordial para el desarrollo de CI/CD debido a su propósito, el cual radica en la estructura, *testing* y despliegue de soluciones.

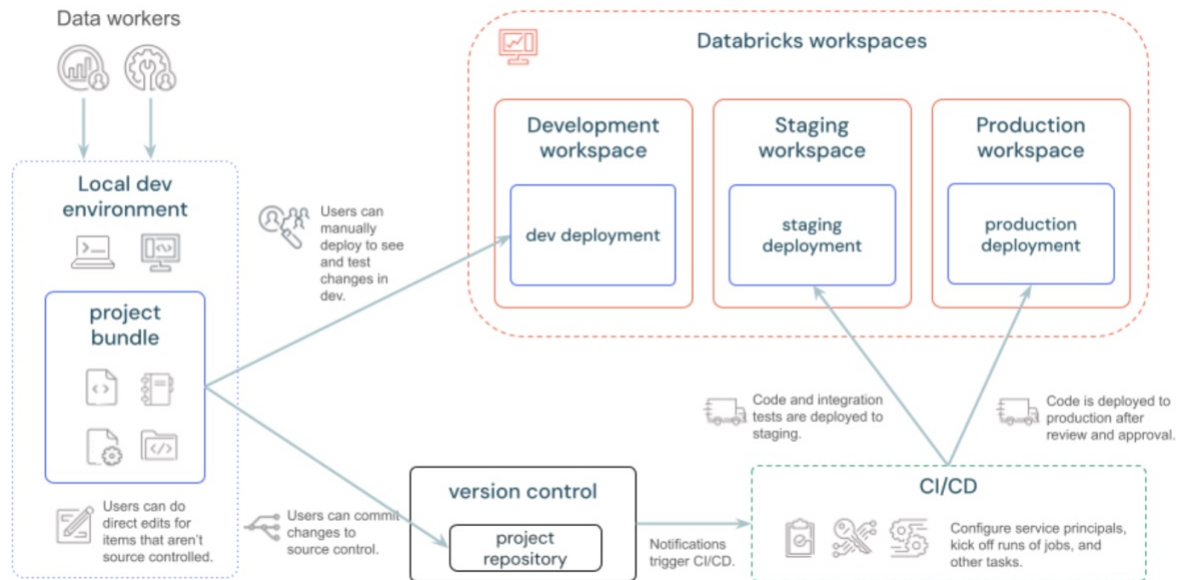
La principal relevancia de esta herramienta dentro del equipo de Ingeniería de Inteligencia Artificial es su capacidad en la formalización de prácticas de CI/CD y su configuración de manera parcial o manual. Según la documentación de Databricks Asset Bundles: “Los Bundles se definen y administran a través de archivos *YAML* que se crean y mantienen junto el código fuente, lo que permite describir recursos de Databricks como archivos de código” (Databricks, 2025). Se puede verificar que DAB permite que los proyectos sean definidos como Infrastructure as Code, mediante archivos *YAML*, los cuales pueden ser desplegados y validados como una interfaz, cumpliendo el propósito de CI/CD.

El funcionamiento de los *pipelines* en DAB es fundamental para la correcta implementación de CI/CD. Este mismo se encuentra estructurado para incluir los archivos según el desarrollo del usuario, donde, mediante los metadatos *YAML*, se define cuáles recursos y componentes serán desplegados mediante Databricks Asset Bundles. A partir de este punto, cada *bundle* puede ser validado mediante el Databricks CLI e incluso su integración con GitHub Actions son primordiales para la creación de *pipelines* automáticos para cualquier tipo de ambiente.

La siguiente imagen refleja el flujo de trabajo de CI/CD utilizando Databricks Asset Bundles:

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Figura 15: Vista a alto nivel de CI/CD pipelines con bundles



Fuente: Databricks Documentation (2025).

Siguiendo este flujo de trabajo, el equipo de Ingeniería de Inteligencia Artificial utiliza esta herramienta como experimentación, sin embargo, el flujo de trabajo del equipo, por lo general, se encuentra en *notebooks* de Databricks, siendo un método más informal y complicado de desplegar en producción.

5. Propuesta de Solución

5.1. Fase 2: Diseño del marco de trabajo

La estandarización del marco de trabajo del equipo de Ingeniería de Inteligencia Artificial no solo requiere de un enfoque técnico adecuado, sino también de la selección cuidadosa de herramientas y procedimientos que respondan a las necesidades operativas del equipo. Principalmente, el equipo del marco de trabajo debe responder a la pregunta: ¿cuál es la visión del equipo y qué aspectos técnicos busca cumplir?

Esta sección tiene como propósito establecer las bases tecnológicas y metodológicas que son clave de la propuesta del marco de trabajo basado en DevOps CI/CD y que se encuentren alineadas con los hallazgos de la fase uno.

La propuesta que se presenta, a continuación, toma como fundamento los principios de DevOps, CI/CD, Lean y AI Readiness Criteria, además, busca integrar soluciones específicas como Databricks Asset Bundles, GitHub, SonarQube y Confluence en una arquitectura coherente. Cada herramienta seleccionada es respuesta a una necesidad existente diagnosticada previamente y su integración en el flujo de trabajo es una respuesta al desarrollo e implementación correcta del marco de trabajo en el ciclo de vida de proyectos del equipo de Ingeniería de Inteligencia Artificial.

5.1.1. Herramientas y procedimientos propuestos según las necesidades del equipo

5.1.1.1. Asociación entre necesidades, herramientas y prácticas

La siguiente tabla resume las principales necesidades identificadas en la fase uno y establece una correspondencia directa entre las herramientas y prácticas propuestas, justificando su integración del marco de trabajo.

Tabla 16: Correlación entre necesidades, herramientas y practicas

Necesidad	Herramienta / procedimiento propuesto	Justificación técnica y funcional
Falta de configuración de CI/CD en los proyectos de la Unidad de Cuidado Familiar.	<i>Databricks Asset Bundles (DABs)</i>	Permiten definir proyectos de Inteligencia Artificial, Ingeniería de datos y de Ciencia de datos como código (<i>IaC</i>)
Procedimientos de despliegue manuales e inconsistentes.	CI/CD automatizado mediante <i>GitHub Actions</i> y <i>DABs</i> .	Se establece la estructura de un <i>pipeline</i> , con control de versiones, donde las validaciones técnicas se integran previo al despliegue. Este mismo evita la intervención manual e incrementa la estabilidad de las versiones.
Los procesos de control de la calidad del código no son consistentes.	SonarQube integrado en el <i>pipeline</i> de CI/CD.	Automatiza el análisis de calidad del código, sin procedimientos manuales adicionales. A su vez, existe la cobertura de pruebas unitarias y vulnerabilidades antes de permitir <i>merges</i> o despliegues.
No se aplican pruebas automatizadas y validaciones técnicas formales.	Integración con el <i>framework pytest</i> mediante la automatización del <i>pipeline</i> .	Inclusión de pruebas unitarias en cada <i>bundle</i> antes de ser desplegado a desarrollo o producción.
Ausencia de trazabilidad completa del desarrollo.	Integración con <i>GitHub</i> e implementación mediante <i>Gitflow</i> o en su defecto una estrategia de <i>branching</i> .	Estructura de trabajo colaborativo que permite la visualización de cada cambio desde el origen hasta la producción. Este mismo permite la revisión de código y control de cambios.

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Necesidad	Herramienta / procedimiento propuesto	Justificación técnica y funcional.
Dificultad para mantener flujos en nuevos proyectos.	Personalización de plantillas mediante DAB.	Facilitan la creación de <i>Bundles</i> y, por ende, la implementación de CI/CD.
Documentación técnica inconsistente o inexistente.	Confluence integrado al proceso, más documentación técnica basada en el marco de operacionalización del equipo de SRE.	Centralización de información técnica estructurada, incluyendo arquitectura, datos, componentes, procesos y dependencias.
Inexistencia del proceso de <i>Code Review</i> .	Se recomienda la implementación de Code Review mediante GitHub Copilot cuando la integración de código no sea al <i>main Branch</i> . Sin embargo, cuando la integración en producción es necesario del proceso de <i>code review</i> .	El <i>code review</i> se vuelve fundamental para cumplir con los estándares esperados de DevOps CI/CD, siendo vital para mantener el flujo de retroalimentación constante.
Incumplimiento de las métricas del AI Readiness Criteria.	Integración del marco de trabajo basado en DevOps CI/CD, el cual permite validar los requerimientos técnicos y operativos para lograr la completitud de las soluciones del equipo de Ingeniería de Inteligencia Artificial.	La aplicación de un marco y el seguimiento de sus métricas vuelven al equipo en uno fiable que presenta soluciones de calidad mediante entregas rápidas y validadas.

Fuente: elaboración propia (2025)

5.1.1.2. Integración de Databricks Asset Bundles en el Marco DevOps CI/CD

La integración de Databricks Asset Bundles constituye una pieza fundamental dentro del marco de trabajo DevOps CI/CD para el equipo de Ingeniería de Inteligencia Artificial. A continuación, se brinda un análisis de por qué la integración de DAB permite cumplir con los requerimientos del equipo de Ingeniería de Inteligencia Artificial.

5.1.1.2.1. Justificación de la herramienta

Actualmente, en la empresa manufacturera donde se realiza el proyecto, se cuenta con una licencia de Databricks, el cual es utilizado a nivel organizacional para la gestión, exploración y administración de datos que provienen del CDL de la compañía y, por ende, es donde se tiene acceso a la información referente a la Unidad de Cuidado Familiar. Por esta razón, es necesaria una herramienta que pueda conectarse directamente al clúster de Databricks y pueda automatizar los procesos manuales que se realizan en la plataforma.

Según la documentación oficial de Databricks Asset Bundles: “*Databricks Asset Bundles* es una herramienta diseñada para facilitar la adopción de las mejores prácticas de ingeniería de software, incluyendo el control de versiones, revisión de código, pruebas y la integración y entrega continua (CI/CD), en proyectos de datos e Inteligencia Artificial” (Microsoft, 2025). Con base en el conjunto de Databricks Asset Bundles, se valora como la mejor opción para la integración del AI Readiness Criteria dentro de las soluciones del equipo, siendo una herramienta realmente útil para el desarrollo de los proyectos a nivel de datos e inteligencia artificial.

En el marco DevOps CI/CD, es necesario contar con una herramienta, plantilla y metodología que permita la integración con Databricks y, a su vez, que esta herramienta siga las pautas del AI Readiness Criteria de la organización. Por esta razón, Databricks Asset Bundles es fundamental en la estandarización del marco de trabajo para los proyectos del equipo de Ingeniería de Inteligencia Artificial.

5.1.1.2.2. Importancia de la herramienta según el marco DevOps CI/CD

Databricks Asset Bundles es esencial para la estandarización y control de flujos de despliegue en ambiente como en el caso del Launchpad. La herramienta permite la ejecución de *pipelines* automatizados, configuración de ambientes y la integración de CI/CD mediante GitHub Actions, siendo ideal para la aplicación de procedimientos con SonarQube o Pytest.

Como señala la documentación oficial de Databricks: “*Databricks Asset Bundles* se utiliza cuando se desea administrar proyectos complejos con múltiples colaboradores y la automatización es esencial mediante el requisito de CI/CD” (Microsoft, 2025). Esta característica convierte a DAB en un componente ideal para equipos como el de Ingeniería de Inteligencia Artificial.

5.1.1.3. Integración de CI/CD mediante GitHub Actions

En la organización donde se desarrolla este proyecto, la utilización de GitHub es un estándar empresarial. Cabe recalcar que la organización de Ingeniería de Inteligencia Artificial cuenta con una licencia de GitHub por cada colaborador que utiliza dicha plataforma. Por esta razón, es la plataforma por defecto.

El uso de GitHub como plataforma central de colaboración consolida ciertos requerimientos expuestos en el AI Readiness Criteria de la organización, siendo requerimientos necesarios para cumplir con el nivel de completitud de métricas de calidad de este mismo.

Dentro de este entorno, GitHub Actions se selecciona como componente fundamental para la realización del *pipeline* CI/CD, debido a que esta herramienta proporciona la automatización basada en eventos como *commits*, *pull requests* o *merges*, siendo fundamental para contar con un Gitflow Strategy, el cual es un componente clave para desarrollar un flujo DevOps CI/CD correcto.

Según Kinsman et al. (2021): “*GitHub Actions*” permite automatizar tareas en función de diferentes *triggers* (por ejemplo, *commits*, *pull requests*, *issues* o *merges*), facilitando la forma en que los desarrolladores construyen, prueban y despliegan proyectos de software”. Esta funcionalidad es necesaria para el desarrollo de un marco de trabajo DevOps CI/CD, constituida por diferentes acciones que permiten la integración de pruebas unitarias, monitoreo, integración continua y el despliegue continuo. Siendo una herramienta fundamental para la integración de distintas tecnologías, tales como SonarQube, Black, Pytest y, sobre todo, Databricks Asset Bundles, permite el despliegue continuo en esta plataforma.

Esta afirmación se refuerza en el análisis de Fairbanks et al. (2023), quienes concluyen que: “En promedio, la implementación de *CI/CD* resulta en una mejora significativa de la velocidad de *commits* del 141.19%, lo que respalda el objetivo principal de esta práctica: acelerar el ciclo de desarrollo mediante la integración y entrega continua”.

GitHub en conjunto con su integración de GitHub Actions responde al estándar tecnológico que busca la organización, habilitando el desarrollo de un *pipeline* CI/CD, contribuyendo al cumplimiento del AI Readiness Criteria y la integración de herramientas esenciales para el monitoreo, validación y despliegue de soluciones.

5.1.1.4. Integración de SonarQube en el Marco DevOps CI/CD

Dentro de un marco DevOps CI/CD que busca la gestión efectiva de proyectos de Ingeniería de Inteligencia Artificial, es fundamental contar con herramientas que aseguren la calidad del código desde las primeras etapas del ciclo de desarrollo. En este sentido, SonarQube se posiciona como una herramienta clave para la realización de análisis estáticos automatizados, identificación de errores, vulnerabilidades de seguridad y problemas de mantenimiento en los despliegues de CI/CD.

Actualmente, la empresa manufacturera cuenta con una licencia de SonarQube, la cual se basa en la cantidad de líneas de código que se pueden evaluar, ya sea por equipo o por trabajador. Facilitando, de esta forma, la herramienta del análisis de código estático.

Se considera que SonarQube es ideal para este marco de trabajo debido a su aplicación con CI/CD y GitHub Actions. Esto debido a que cuenta con los parámetros necesarios para el análisis estático de código. Según la tesis de análisis de seguridad y calidad de aplicaciones: “El análisis estático es un proceso que se realiza sobre el código de una aplicación sin la necesidad de ejecutarse. Representa un análisis realizado de forma automatizada por una herramienta, donde su idea principal es encontrar tan pronto como sea posible, defectos de codificación en el ciclo de vida del desarrollo” (Ospina Delgado, 2015).

Es correcto afirmar que la aplicación de SonarQube como herramienta de análisis de la calidad de código es realmente compatible con el propósito del marco de trabajo DevOps CI/CD, estableciendo integración mediante CI/CD y un análisis completo del funcionamiento real de la aplicación en etapas de desarrollo, alineándose con los parámetros de AI Readiness Criteria.

5.1.1.5. Integración de pruebas unitarias mediante *pytest* en el marco DevOps CI/CD

En el marco de desarrollo DevOps CI/CD propuesto, la integración de pruebas unitarias automatizadas es un requisito indispensable en los lineamientos de la organización de Ingeniería de Inteligencia Artificial. Este procedimiento es esencial para la validación continua mediante procedimientos de CI/CD, reducción de errores y generación de ciclos de retroalimentación.

Según Okken (2022): “*pytest* es una herramienta robusta para *Python* que puede utilizarse para todo tipo y nivel de pruebas de software. Se integra fácilmente con herramientas como integración continua y automatización”. Este *framework* se escoge por su mantenibilidad a nivel de CI/CD, adoptando funcionalidades de cobertura de archivos y líneas de código. Su nivel de integración con GitHub Actions hace a la herramienta ideal para la gestión mediante SonarQube y su despliegue próximo.

Pytest ofrece compatibilidad con diferentes niveles de soporte, *plugin* y *unittest*, lo cual permite establecer pruebas versionadas y automatizadas en cada despliegue realizado. Esto facilita el cumplimiento del AI Readiness Criteria asegurando que cada componente del *pipeline* sea evaluado automáticamente en cuanto a seguridad y cobertura.

Es correcto afirmar que el *pytest* cumple un rol clave dentro del marco DevOps CI/CD al facilitar la integración mediante GitHub Actions, estándares de calidad y completitud a nivel de proyectos de inteligencia artificial.

5.1.1.6. Integración de Code Review en el Marco DevOps CI/CD

El *code review* es un lineamiento importante para la integración de DevOps y métricas de calidad. Este procedimiento es necesario para el cumplimiento de lineamientos del AI Readiness Criteria.

Actualmente, se han implementado acciones dentro de GitHub para el *code review* automático de diferentes *branches* de desarrollo, por esta razón, esta herramienta puede ser realmente funcional para la revisión progresiva de procedimientos de integración continua de *software*. En la prueba de concepto como tal, se visualiza la integración de esta práctica.

Es necesario habilitar funcionalidades de *code review* colaborativo, donde sea necesaria la verificación de código y el líder técnico del proyecto sea el encargado de aprobar los cambios a un estado de producción. Esto con el fin de adoptar una práctica efectiva para el despliegue continuo, en el cual el *Gitflow* no sea completamente dependiente de una persona que revise los cambios según cada *feature*, sino que *Copilot* sea una herramienta que agilice este proyecto y, posteriormente, un líder técnico sea el encargado de aprobar los cambios para producción.

5.1.2. Especificaciones administrativas, técnicas y operativas del marco DevOps CI/CD

5.1.2.1. Fase de planificación

La fase de planificación busca alinear todos los requerimientos técnicos, de negocio y de completitud para el desarrollo efectivo del marco de trabajo. A continuación, se describen las etapas en las cuales se desarrolla la fase de planificación, y cómo estas metodologías y procesos se ven ejemplificadas en el ciclo de vida de los proyectos del equipo.

Es necesario destacar que se realiza la adopción de metodologías ágiles, tomando en cuenta historias de usuario, herramientas como Jira y Confluence para el desarrollo de la planificación.

Tabla 17: Especificaciones del marco de trabajo en la fase de planificación

Metodología aplicada	Procedimiento aplicable al equipo de Ingeniería de Inteligencia Artificial.
Lean	<u>Visualización del trabajo en curso</u> <ul style="list-style-type: none">• Para la recopilación de requerimientos y gestión de estos mismos, se maneja desde Confluence.• La incorporación de Kanban Boards es necesaria para la identificación de prioridades, bloqueos y estado de las tareas.
	<u>Trabajar en pequeños lotes</u> <ul style="list-style-type: none">• Implicación de una metodología mediante cambios pequeños, revisiones frecuentes y semanales. Este principio se desea implementar según los requerimientos del negocio, sin embargo, lo ideal es que la retroalimentación sea constante.• Establecimiento de límites de WIP mediante el <i>product owner</i> dedicado del proyecto. Efectuando una priorización de requerimientos mediante importancia, tiempo e impacto, estableciendo el número de tareas que se desarrollan en ese momento por parte del ingeniero.• Establecimiento del <i>Lead Time</i> esperado por el <i>product owner</i>, con base en el ciclo de vida del proyecto.

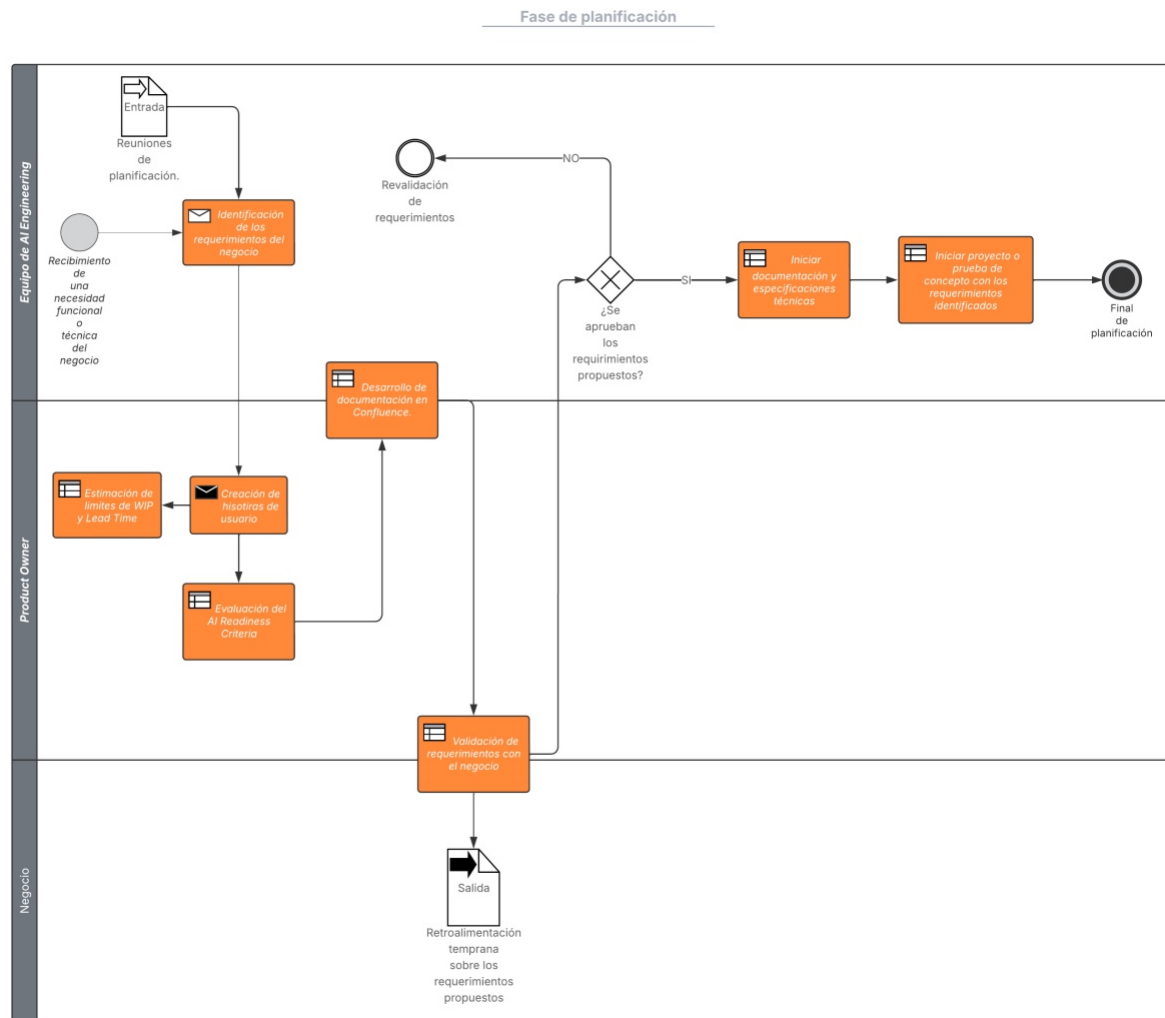
Metodología aplicada	Procedimiento aplicable al equipo de Ingeniería de Inteligencia Artificial.
<i>DevOps</i>	<p><u>Flujo DevOps</u> Visualización E2E del proceso desde la planificación hasta el despliegue. De esta forma, se define el flujo de valor mediante diagramas BPMN 2.0.</p> <hr/> <p><u>Cultura de mejora continua</u></p> <ul style="list-style-type: none"> • Retrospectivas tempranas sobre el ciclo de vida de los proyectos. • Involucramiento del equipo encargado del proyecto incluyendo <i>AI Engineers</i>, <i>Product Owners</i> y <i>Stakeholders</i> del negocio. Si el proyecto y la planificación lo permiten, también debe integrarse el equipo de operaciones o SRE.
<i>AI Readiness Criteria</i>	<p><u>Evaluación inicial de métricas</u></p> <ul style="list-style-type: none"> • Revisión de estándares de documentación, pruebas, seguridad y seguimiento constante de la lista de requerimientos del AI Readiness Criteria en el proyecto • Identificación de <i>gaps</i> para el desarrollo del proyecto.
Documentación	<p><u>Documentación técnica</u></p> <ul style="list-style-type: none"> • Documentación del modelo de soporte del proyecto, incluyendo quienes son los <i>Stakeholders</i>, <i>Product Owner</i>, <i>AI Engineers</i>, ingenieros de datos y los que sean necesarios. • Es necesario tomar en cuenta que por lo general es el <i>AI Engineer</i> el <i>product owner</i> del proyecto, o en su defecto el contacto directo con la unidad de cuidado familiar. • Diagrama de infraestructura, <i>dataflow</i>, componentes iniciales y dependencias. • Modelo de flujo de trabajo. • Documentación de requerimientos validados por el negocio y creación inicial de historias de usuario. • Documentación de accesos necesarios para el acceso a la información. • Documentación del nombre, repositorio, ambientes y plantilla de transición a producción.

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Fuente: elaboración propia (2025).

5.1.2.1.1. Flujo de la fase de planificación

Figura 16: Flujo de la fase de planificación



Fuente: elaboración propia (2025).

5.1.2.2. Fase de desarrollo y de *testing*.

La fase de desarrollo y testing es esencial dentro del ciclo DevOps CI/CD, ya que integra los principios de automatización, control de calidad y la colaboración entre el desarrollo de soluciones informáticas, seguridad y operaciones.

Tabla 18: Especificaciones del marco de trabajo en la fase de desarrollo y testing

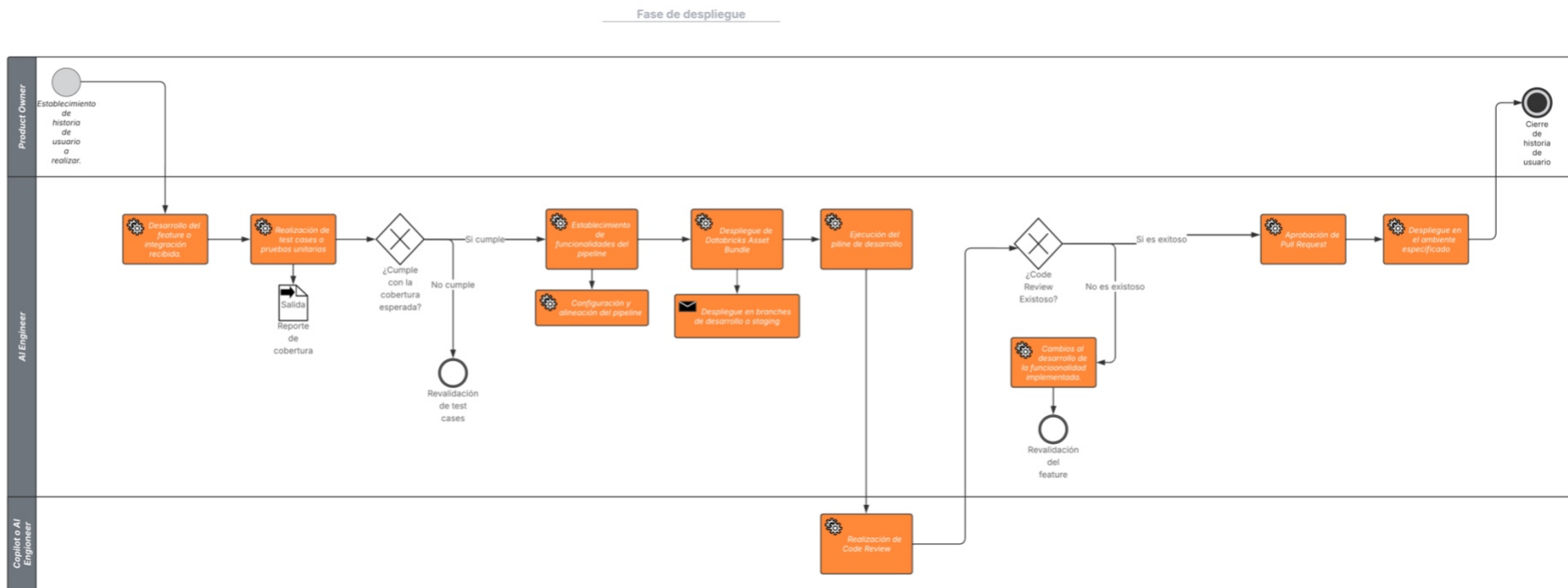
Metodología aplicada	Procedimiento aplicable al equipo de Ingeniería de Inteligencia Artificial.
<i>Lean</i>	<p><u>Gestión del flujo de valor</u></p> <ul style="list-style-type: none"> • Todo <i>feature</i> o cambio debe estar asociado a una historia de usuario. • El trabajo se organiza en ciclos cortos, como entregables funcionales. • Se promueve la integración frecuente de código y despliegues regulares en entornos de desarrollo y pruebas.
<i>DevOps</i>	<p><u>Automatización y colaboración</u></p> <ul style="list-style-type: none"> • Se adopta un enfoque <i>trunk-based</i> o <i>GitFlow</i> definido previamente en la fase de planificación. Este proceso ejemplifica qué tipo de integración se está realizando, con <i>branches</i> tales como <i>develop</i>, <i>feature/</i>, <i>main</i> o <i>hotfix/</i>. • La integración de código activa <i>pipelines</i> que validan aspectos del código en función al cumplimiento de calidad, seguridad y pruebas unitarias.
	<p><u>Gestión de Code Review.</u></p> <ul style="list-style-type: none"> • El <i>code review</i> debe ser aplicado en cualquier integración de <i>branch</i>, sin embargo, esta revisión de código se debe realizar mediante Copilot. • Cuando se gestiona el <i>code review</i> para un despliegue de producción, es necesario contar con el proceso de aprobación de código de un líder técnico.
<i>CI/CD</i>	<p><u>Continuous Integration and Continuous Deployment</u></p> <ul style="list-style-type: none"> • Todo <i>pull request</i> activa el <i>pipeline</i> en GitHub Actions. • Se ejecutan tareas automatizadas: <ul style="list-style-type: none"> ○ Organización de código (<i>isort</i>, <i>black</i>). ○ Pruebas unitarias (<i>pytest</i>). ○ Cobertura de pruebas unitarias. ○ Escaneo de calidad de código con SonarQube. • La integración se establece en el <i>pipeline</i> mediante la configuración de comandos de Databricks Asset Bundles.

Metodología aplicada	Procedimiento aplicable al equipo de Ingeniería de Inteligencia Artificial.
Documentación	<u>Especificaciones técnicas</u> <ul style="list-style-type: none">• El código debe estar versionado en GitHub con proceso de revisión de código.• Se den de incluir <i>test cases</i> y resultados.• Documentación de la estrategia de CI/CD a utilizar.• Documentación de todos los GitHub secrets dentro del <i>pipeline</i>.
<i>AI Readiness Criteria</i>	<u>Cumplimiento técnico-operativo</u> <ul style="list-style-type: none">• Cumplimiento de los procesos de seguridad definidos por IRisk.• Documentación actualizada de dependencias, datasets, lógica de negocio y funciones de inferencia. Inclusión de métricas de validación del <i>pipeline</i> .

Fuente: elaboración propia (2025).

5.1.2.2.1. Flujo de la fase de desarrollo y testing

Figura 17: Flujo de la fase de desarrollo y testing



Fuente: elaboración propia (2025).

5.1.2.3. Fase de despliegue y monitoreo

La fase de despliegue y monitoreo corresponde al cierre técnico o la última fase del ciclo DevOps CI/CD, en la cual se realiza el despliegue final en el entorno de producción y se establecen mecanismos formales de monitoreo, validación y soporte en operaciones.

Tabla 19: Especificaciones del marco de trabajo en la fase de despliegue y monitoreo

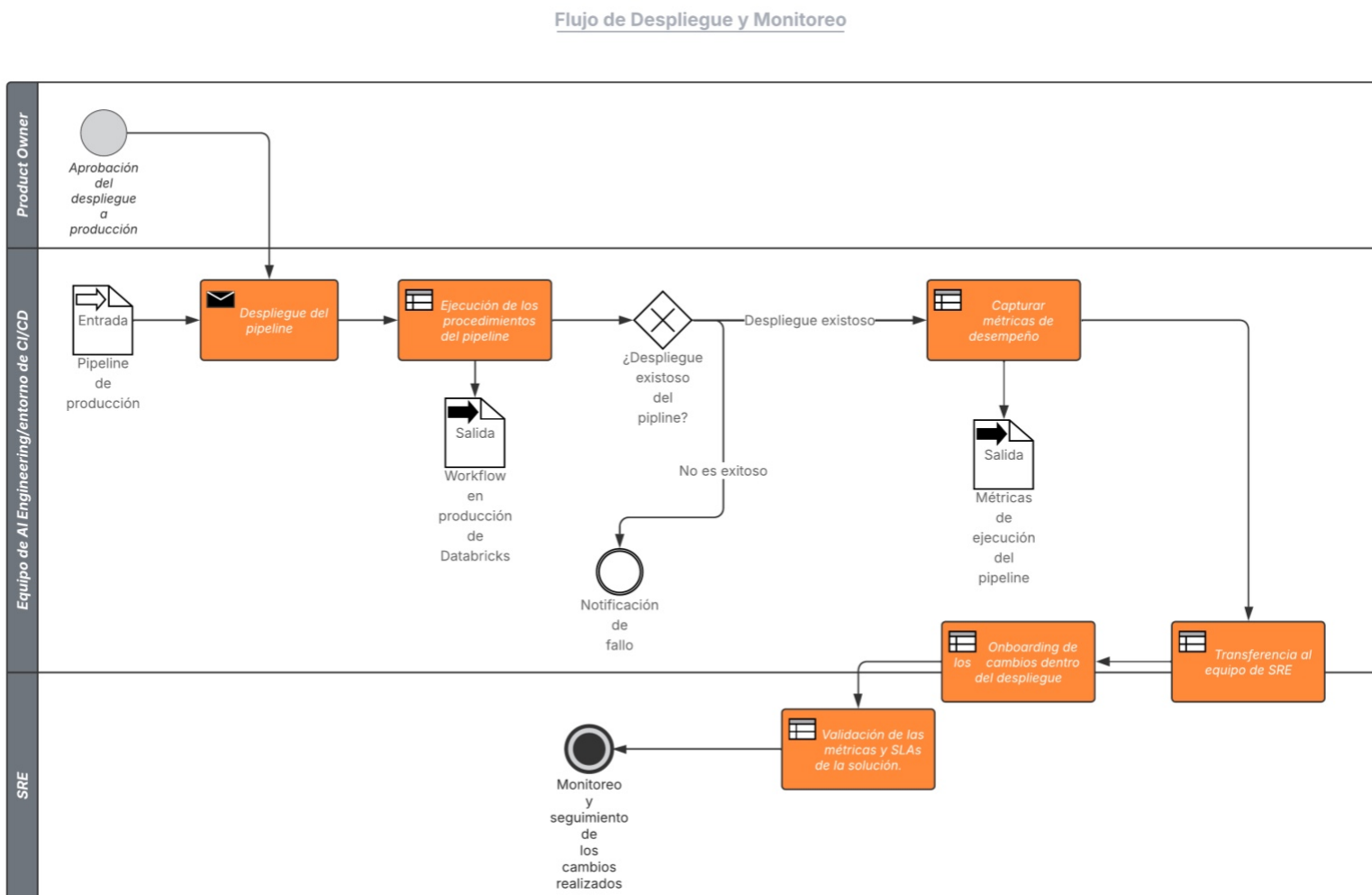
Metodología aplicada	Procedimiento aplicable al equipo de Ingeniería de Inteligencia Artificial.
Lean	<p><u>Minimización del desperdicio y generación de valor</u></p> <ul style="list-style-type: none"> • Se automatiza el despliegue a producción para reducir tiempos muertos, este debe ser consistente con el <i>Lead Time</i> planteado en la planificación. • Se registran métricas de tiempo, disponibilidad, estabilidad y errores en el despliegue de soluciones. • Monitoreo continuo mediante los <i>workflow</i> desplegados en Databricks.
DevOps	<p><u>Transferencia a operaciones y monitoreo colaborativo</u></p> <ul style="list-style-type: none"> • Entrega de documentación al equipo de SRE. • Se notifica la implementación del nuevo <i>feature</i> desarrollado. Además, se brinda <i>onboarding</i> del funcionamiento de este. • Se validan o se desarrollan los SLA que se implican en la aplicación según el momento en el que se realiza dicha integración.
CI/CD	<p><u>Despliegue automatizado y monitoreo post – despliegue</u></p> <ul style="list-style-type: none"> • El paso a producción se ejecuta desde un <i>pull request</i> al <i>main branch</i> mediante la integración con GitHub Actions y el ambiente de producción de Databricks. • Se realizan los pasos de validación descritos en la validación del <i>pipeline</i> y desarrollo, como gestión de la calidad del código y pruebas automatizadas.
Documentación	<ul style="list-style-type: none"> • Documentación de la ejecución del procedimiento. • Flujo desarrollado. • Configuración de alarmas, <i>dashboards</i> y seguimiento mediante el equipo de SRE. • Validación con el negocio de los requerimientos y <i>features</i> integrados.

Metodología aplicada	Procedimiento aplicable al equipo de Ingeniería de Inteligencia Artificial.
AI Readiness Criteria	<u>Cumplimiento técnico-funcional</u> <ul style="list-style-type: none">• Validación de que la solución desplegada cumple con los requisitos y métricas definidas en la etapa de planificación en implicación del AI Readiness Criteria.• Inclusión de métricas de monitoreo, herramientas, seguridad, modelos y rendimiento de la aplicación.
<i>DevOps</i>	<u>Transferencia a operaciones y monitoreo colaborativo</u> <ul style="list-style-type: none">• Entrega de documentación al equipo de SRE.• Se notifica la implementación del nuevo <i>feature</i> desarrollado. Además, se brinda <i>onboarding</i> del funcionamiento de este.• Se validan o se desarrollan los SLA que se implican en la aplicación según el momento en el que se realiza dicha integración.

Fuente: elaboración propia (2025).

5.1.2.3.1. Flujo de la fase de despliegue y monitoreo

Figura 18: Flujo de la fase de despliegue y monitoreo



Fuente: elaboración propia (2025).

5.1.2.4. Desarrollo del *pipeline*

Para la definición operativa y técnica del marco de trabajo, se establece una estructura estándar de *pipeline* CI/CD, diseñado para los trabajos del equipo de Ingeniería de Inteligencia Artificial en la Unidad de Cuidado Familiar. El enfoque de esta metodología es desarrollar la cultura de CI/CD para los proyectos de esta organización, adoptando las buenas prácticas, herramientas y tecnologías que estas conllevan. A continuación, se describen las fases y componentes del *pipeline*:

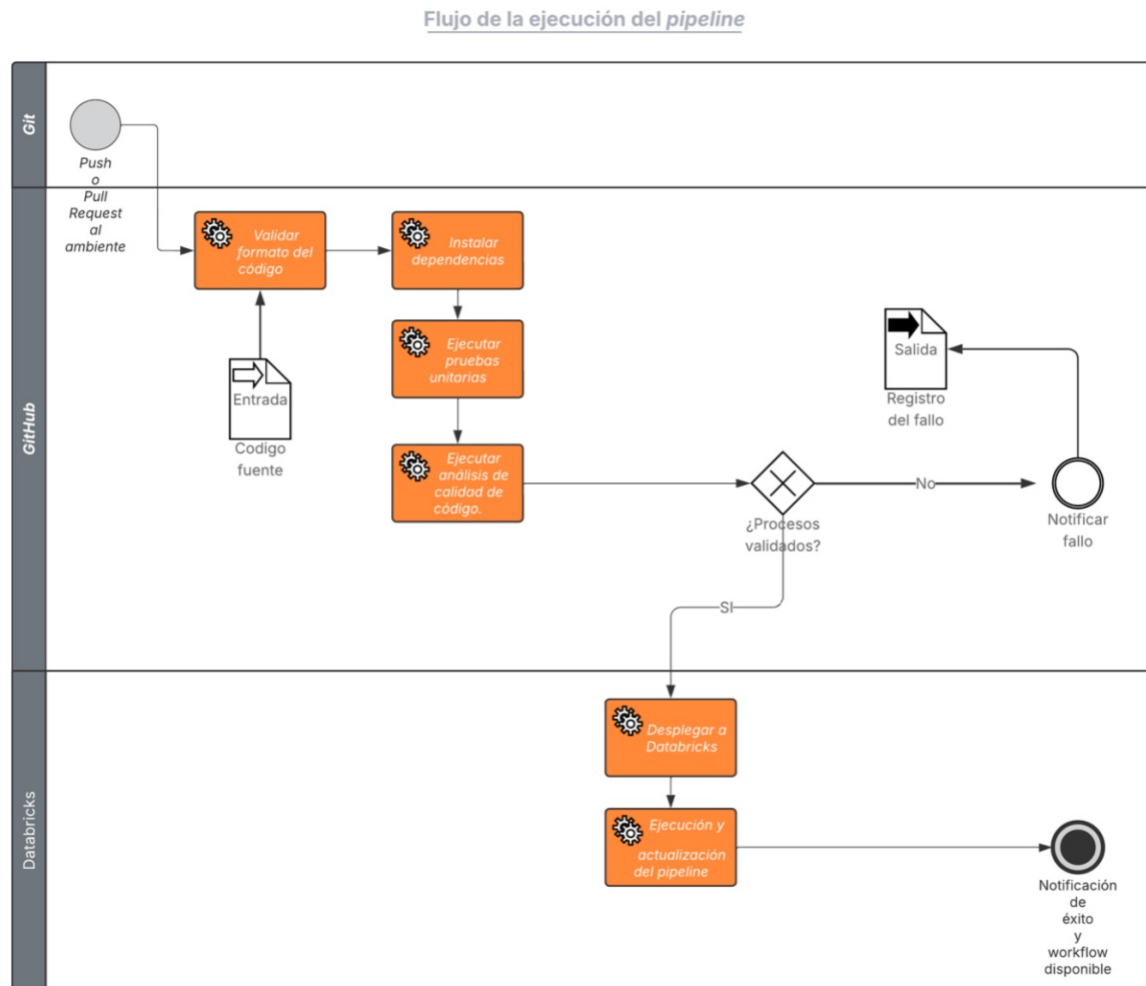
Figura 19: Fases del desarrollo del *pipeline* para el marco propuesto

Fases del <i>pipeline</i>	Descripción	Pasos del <i>pipeline</i>
Validación del procedimiento y pruebas de calidad.	Dependiendo del ambiente en el que se desarrolla, este <i>job</i> se activa ante eventos <i>push</i> o <i>pull request</i> , ya sea en ramas para <i>features</i> , <i>hot fix</i> , <i>releases</i> , <i>develop</i> o <i>main</i> . Esto habilita una política de validación previa a merge.	<ol style="list-style-type: none"> 1. Instalación de dependencias y entorno virtual. 2. Análisis de formato de código usando herramientas como <i>black</i> e <i>isort</i>. 3. Ejecución de pruebas unitarias con <i>pytest</i>, incluyendo la generación de reporte de cobertura (<i>coverage.xml</i>). 4. Posteo del reporte de cobertura para análisis externo. 5. Ejecución de SonarQube, con reglas de calidad del código, análisis de seguridad y revisión de pruebas.
Despliegue del código en el ambiente especificado.	Una vez completada la etapa de validación, el <i>pipeline</i> realiza el despliegue del <i>bundle</i> al entorno de Databricks.	<ol style="list-style-type: none"> 1. El proceso replica la infraestructura, <i>notebooks</i>, tareas y configuraciones definidas en el archivo fuente de <i>databrick.yml</i>. 2. El entorno de despliegue se controla mediante variables de entornos.
Ejecución del bundle en el ambiente.	El tercer <i>job</i> ejecuta la lógica del bundle ya desplegado, mediante comandos de Databricks.	<ol style="list-style-type: none"> 1. Se realiza la validación de que el <i>job</i> esta correctamente definido. 2. Se valida que los recursos y configuraciones del entorno estén activos. 3. Se valida que el flujo de datos funcione según lo esperado, esto se puede complementar visualizando el <i>workflow</i> de Databricks.

Fuente: elaboración propia (2025).

5.1.2.4.1. Flujo del pipeline

Figura 20: Flujo de ejecución del Pipeline



Fuente: elaboración propia (2025).

5.2. Fase 3: Comparación de la situación actual versus Propuesta de marco de trabajo para la Gestión de los proyectos de Ingeniería de Inteligencia Artificial.

5.2.1. Procesos modificados e integrados al pipeline de desarrollo

Como resultado del análisis de la situación actual del equipo de Ingeniería de Inteligencia Artificial, se identifica que los procesos relacionados con la planificación, desarrollo, *testing* y despliegue de los proyectos carecían de una estructurada clara y que fuera un estándar para la realización de estos en la Unidad de Cuidado Familiar.

Con la propuesta del marco de trabajo basado en DevOps CI/CD, se diseñó e integró un conjunto de procesos, metodologías y métricas para garantizar la calidad técnica y operativa en todas las fases del ciclo de vida de proyectos según DevOps, abarcando desde la planificación hasta el despliegue de estos. De esta forma, se respalda la calidad técnica, automatización y el cumplimiento normativo de las soluciones informáticas desarrolladas por el equipo de Ingeniería de Inteligencia Artificial. A continuación, se detallan los principales procesos modificados e integrados en el flujo de trabajo del equipo con la propuesta desarrollada:

Tabla 20: Tabla comparativa de la situación actual versus Procesos modificados

Elemento del proceso	Situación actual	Propuesta de estandarización.
Recopilación de requerimientos	Informal sin estructura documental definida.	Documentación estructurada en Confluence con plantilla base para la gestión de historias de usuario.
Validación técnica	No se realiza	Se evalúa el marco del AI Readiness Criteria.
Estructura de desarrollo	Cada ingeniero define su estructura y entorno manualmente.	Estándar de Databricks Asset Bundles con posibilidad a experimentación, control de versiones y estructura unificada.
Control de versiones	Sin uso necesario de repositorios o estrategia de <i>branches</i> .	Gitflow adaptado con branches para cada tipo de integración, así como el uso de GitHub para la gestión del código.
Integración de código	Sin pruebas automatizadas ni <i>code review</i>	<i>Pull requests</i> obligatorios con integración automatizada de SonarQube y Pytest.

Elemento del proceso	Situación actual	Propuesta de estandarización
<i>Code Review</i>	Opcional o realizada informalmente, sin un proceso formal.	Proceso formal de revisión técnica asistida por GitHub Copilot y validación previa.
Pruebas unitarias	No existe proceso formal para su implementación.	Automatización con <i>pytest</i> , integrando cobertura validada del <i>pipeline</i> .
Validación de calidad y seguridad	No existe integración de este tipo de herramientas en el flujo actual.	Integración mediante SonarQube automatizado en el <i>pipeline</i> y seguimiento mediante la estrategia de <i>branches</i> . A su vez, se mantiene control de secretos y validación mediante GitHub Actions.
Despliegue a producción	No existe una diferenciación entre ambientes de desarrollo y producción.	Se busca cumplir el ciclo DevOps mediante la integración de un ambiente de desarrollo y uno de producción. A su vez, esta implementación se realiza mediante el uso de un <i>pipeline</i> automatizado.
Monitoreo de la ejecución de la solución informática.	Se monitorea únicamente que el producto final se encuentre en el estado adecuado.	Métricas, logs y seguimiento mediante Confluence, Jira, Databricks y GitHub Actions.
Transferencia al equipo de operaciones (<i>SRE</i>)	No formalizada.	Documentación de técnica validada, con los procedimientos de la solución informática, datos, métricas y procesos que facilitan la integración con el equipo de SRE.

Fuente: elaboración propia (2025).

5.3. Fase 4: Resultados de la prueba de concepto

5.3.1. Evaluación de éxito de la prueba según el marco de trabajo propuesto

Para validar la efectividad de la prueba de concepto del marco de trabajo de DevOps CI/CD, se implementa un *pipeline* de desarrollo que contiene las siguientes características:

- Desarrollo de un *feature* mediante un Git Branch Strategy, implementando el control de versiones.
- Integración continua de un proceso automatizado que contiene la instalación de las dependencias, ejecución de pruebas unitarias automatizadas, verificación de calidad de código mediante SonarQube, reporte de cobertura y validación de formato del código.
- Posteriormente, se desarrolla el despliegue continuo de las funcionalidades y lógica por medio de CI/CD en conjunto de Databricks Asset Bundles y GitHub Actions.

El éxito de la prueba se define mediante los siguientes criterios.

- Se comparan las métricas del procedimiento actual del equipo con el *pipeline* CI/CD.
- Se validan que los pasos del *pipeline* sean exitosos, desde el testing, análisis y ejecución de *Jobs* en Databricks.
- El proceso debe ser escalable y adaptable a nuevas funcionalidades, ramas de desarrollo y estabilidad del sistema.

Tomando en cuenta estos criterios, es posible concluir que la prueba es exitosa y se demuestran los beneficios esperados del marco de trabajo, incluyendo automatización, reducción de tiempos y estandarización de procedimientos mediante DevOps y CI/CD.

5.3.2. Contexto del proyecto piloto en la Unidad de Cuidado Familiar

En esta sección, se realiza la prueba de concepto, la cual fue implementada en el contexto de un proyecto activo del equipo de Ingeniería de Inteligencia Artificial, enfocado en brindar soporte técnico a una solución conocida como *RCCP (Rough Cut Capacity Planning)*, perteneciente a la Unidad de negocio de Cuidado Familiar.

Esta solución tiene como objetivo principal optimizar la planificación mensual de capacidad manufacturera para los próximos 24 meses, hablando específicamente de producto. De esta forma, se consideran restricciones de producción, inventario y transporte, con el fin de minimizar el costo total asociado a RCCP. Por esta razón, su desarrollo es principalmente enfocado en la ingeniería de datos.

Esta solución informática es utilizada por líderes de negocio y líderes de cadena de suministro, quienes mensualmente evalúan variables como la proyección de demanda,

capacidades de inventario y costos operativos, con el fin de tomar decisiones tales como las siguientes:

- Cuándo y dónde producir qué productos.
- Dónde almacenar y cómo distribuir inventarios.
- Cómo planificar el transporte a plantas manufactureras.

En este contexto, el equipo de Ingeniería de Inteligencia Artificial tiene la responsabilidad de operacionalizar el proyecto, es decir, llevarlo a un estado estable, escalable y listo para su gestión en producción; este mismo alineado al marco propuesto DevOps CI/CD que se visualiza en este proyecto.

Para esta prueba de concepto, se enfocó el trabajo en dos principales flujos específicos de gestión de datos, los cuales forman parte del proceso técnico llamado RCCP. Estos flujos consisten en tareas principalmente de tipo ETL (*Extract, Transform, Load*), los cuales permiten preparar, transformar y estandarizar archivos que posteriormente serán recursos para el modelo RCCP.

El propósito de esta prueba de concepto fue integrar el marco propuesto en estos procesos concretos, integrando especificaciones y procedimientos técnicos de la metodología DevOps CI/CD mediante el desarrollo de las siguientes prácticas:

- Control de versiones con GitHub.
- Automatización y CI/CD mediante GitHub Actions.
- Validación de calidad del código mediante SonarQube.
- Integración de pruebas automatizadas.
- Estructuración de código y despliegue mediante Databricks Asset Bundle.
- Ejecución y monitoreo mediante el *pipeline* CI/CD.

5.3.3. Implementación de especificaciones y procedimientos técnicos DevOps CI/CD

5.3.3.1. *Unit Testing*

Una de las prácticas claves dentro del marco DevOps CIO/CD es la implementación de pruebas automatizadas, específicamente *unit tests*. Estas pruebas permiten verificar que cada componente o función del código se comporte según lo esperado, verificando los diferentes casos en los que esta función puede desarrollarse.

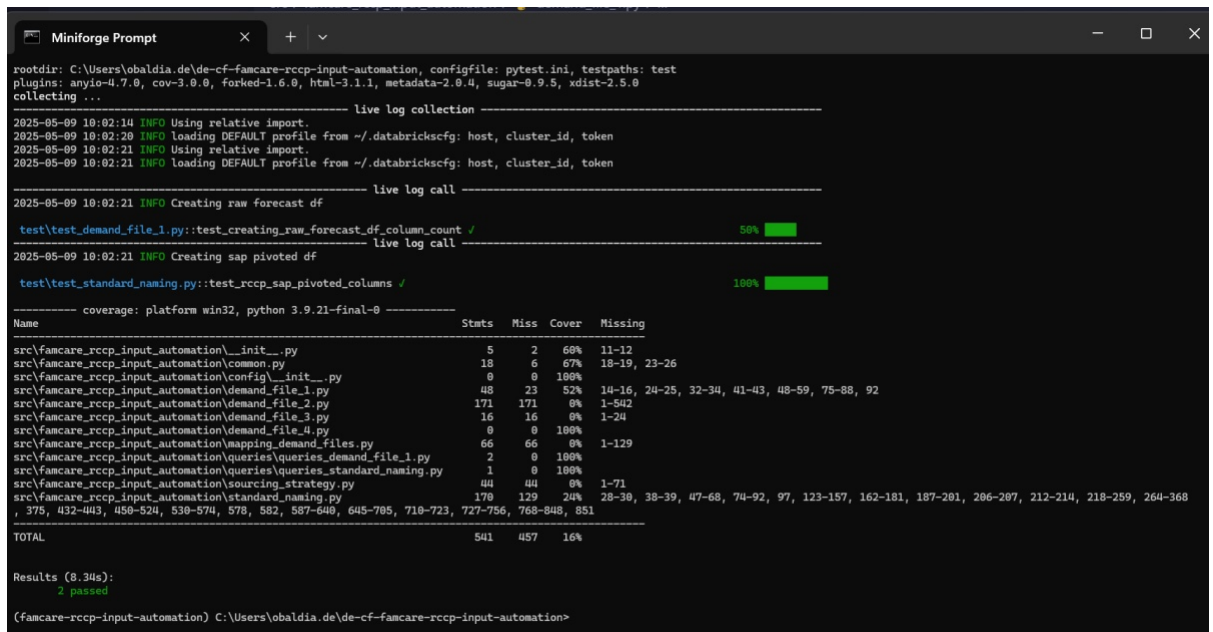
En el contexto de la prueba de concepto, se integran pruebas unitarias automatizadas como parte del *pipeline* de desarrollo. Estas pruebas son construidas mediante el *framework* conocido como *pytest*, el cual es ejecutado automáticamente cada vez que se realiza un *pull request* o un *push* sobre las ramas definidas que, en este caso, corresponden a *develop* y *releases*. Esto se realiza mediante la configuración del archivo conocido como *.github/workflows/dev.yml*, el cual viene integrado con la plantilla de Databricks Asset Bundles.

Como evidencia de esta implementación, se provee la imagen de pruebas unitarias de la prueba de concepto, en la que se puede observar la ejecución completa en el entorno local del usuario. Este procedimiento es repicado en el *pipeline* en el cual se desarrollan los siguientes pasos:

- Desarrollo de dos *test cases* específicos:
 - *test_demand_file_1.py*: verifica el correcto *pivot* de columnas.
 - *test_standard_naming.py*: valida la estandarización del nombre de las columnas.
- Cobertura de código, fundamental para integrarlo a herramientas de control de calidad como SonarQube.
- Los resultados de las pruebas:
 - Un total de tres archivos testados con éxito.
 - Una cobertura global del código del 16%, con desglose por archivo, líneas ejecutadas y líneas omitidas.
 - El resultado general de que las pruebas fueron exitosas.

Este paso del ciclo de desarrollo DevOps CI/CD permite al equipo asegurar que los procesos ETL definidos en esta fase del proyecto no solo funcionan correctamente, sino que son verificables y repetibles en cualquier ejecución del *pipeline*.

Figura 21: Pruebas unitarias de la prueba de concepto



```
Miniforge Prompt
rootdir: C:\Users\obaldia.de\de-cf-famcare-rcpp-input-automation, configfile: pytest.ini, testpaths: test
plugins: anyio-4.7.0, cov-3.0.0, forked-1.6.0, html-3.1.1, metadata-2.0.4, sugar-0.9.5, xdist-2.5.0
collecting ...
----- Live log collection -----
2025-05-09 10:02:14 INFO Using relative import.
2025-05-09 10:02:20 INFO loading DEFAULT profile from ~/.databrickscfg: host, cluster_id, token
2025-05-09 10:02:21 INFO Using relative import.
2025-05-09 10:02:21 INFO loading DEFAULT profile from ~/.databrickscfg: host, cluster_id, token
----- Live log call -----
2025-05-09 10:02:21 INFO Creating raw forecast df
test/test_demand_file_1.py::test_creating_raw_forecast_df_column_count ✓ 50% ██████████
----- Live log call -----
2025-05-09 10:02:21 INFO Creating sap pivoted df
test/test_standard_naming.py::test_rcpp_sap_pivoted_columns ✓ 100% ██████████
----- coverage: platform win32, python 3.9.21-final-0 -----
Name                                                                    Stmts  Miss  Cover   Missing
-----
src/famcare_rcpp_input_automation/__init__.py                            5      2    60%   11-12
src/famcare_rcpp_input_automation/common.py                             18      6    67%   18-19, 23-26
src/famcare_rcpp_input_automation/config/__init__.py                     0      0   100%
src/famcare_rcpp_input_automation/demand_file_1.py                       48     23    52%   14-16, 24-25, 32-34, 41-43, 48-59, 75-88, 92
src/famcare_rcpp_input_automation/demand_file_2.py                      171    171     0%   1-542
src/famcare_rcpp_input_automation/demand_file_3.py                       16     16     0%   1-24
src/famcare_rcpp_input_automation/demand_file_4.py                       0      0   100%
src/famcare_rcpp_input_automation/mapping_demand_files.py                66     66     0%   1-129
src/famcare_rcpp_input_automation/queries/queries_demand_file_1.py        2      0   100%
src/famcare_rcpp_input_automation/queries/queries_standard_naming.py     1      0   100%
src/famcare_rcpp_input_automation/sourcing_strategy.py                   44     44     0%   1-71
src/famcare_rcpp_input_automation/standard_naming.py                    170    129    24%   28-30, 38-39, 47-68, 74-92, 97, 123-157, 162-181, 187-201, 206-207, 212-214, 218-259, 264-368
, 375, 432-443, 450-524, 530-574, 578, 582, 587-640, 645-705, 710-723, 727-756, 760-840, 851
TOTAL                                                                    541    457    16%

Results (8.34s):
  2 passed
(famcare-rcpp-input-automation) C:\Users\obaldia.de\de-cf-famcare-rcpp-input-automation>
```

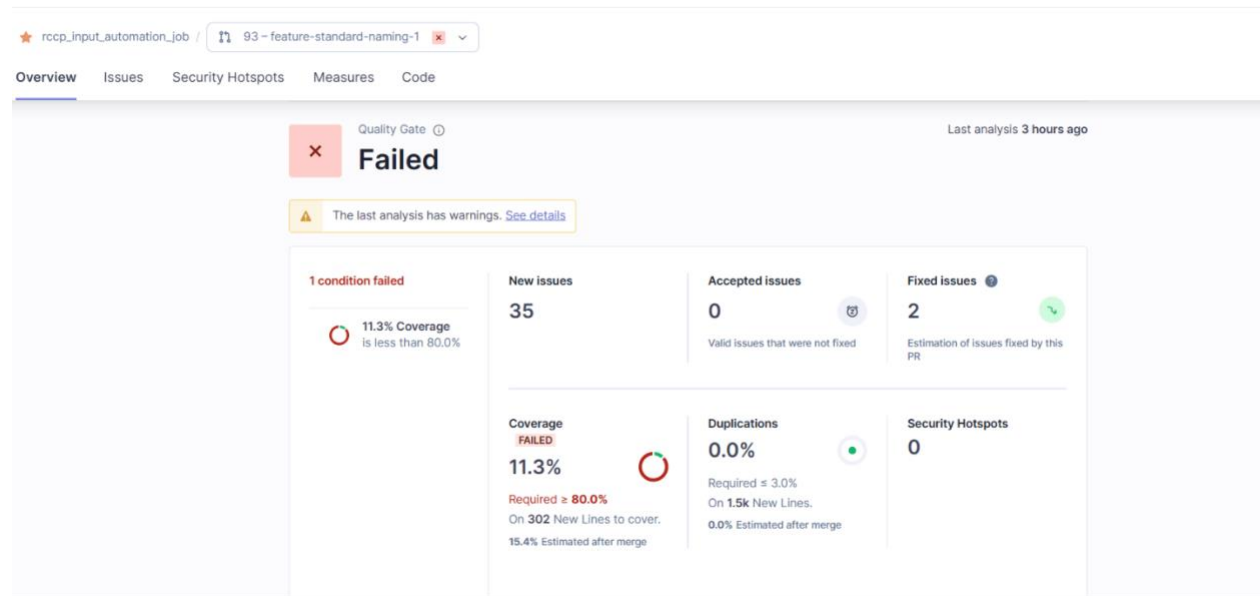
Fuente: elaboración propia (2025).

5.3.3.2. Calidad de código

La evaluación de la calidad de código es una práctica fundamental dentro del marco DevOps CI/CD, esta permite validar de forma automática que el código fuente de la solución entregada cumpla con los estándares de mantenibilidad, simplicidad, cobertura, legibilidad y seguridad. En el marco propuesto, se establece que es necesaria la inclusión de SonarQube para el análisis estático de código dentro del *pipeline* de desarrollo.

La siguiente imagen demuestra el resultado del análisis más reciente para esta prueba de concepto, específicamente del *Branch* llamado “*feature-standard-naming*”. El análisis de este fue clasificado como “fallido”, esto debido a que no alcanzó la cobertura mínima de pruebas establecido por SonarQube, el cual corresponde a 80%.

Figura 22: Visualización de SonarQube en la prueba de concepto



Fuente: elaboración propia (2025).

5.3.3.2.1. Resultados destacados del análisis

- Cobertura de pruebas:
 - 11.3% de cobertura total de todas las líneas.
 - Esto permite verificar los archivos y funciones no validadas de las pruebas realizadas.
- Duplicación de código
 - 0.0% de duplicación de código, lo cual indica una buena práctica en términos de reutilización de funciones.
- Errores críticos (*Bugs*)

- 2 errores corregidos durante la prueba.
- Nuevas observaciones
 - 35 *issues* menores correspondientes a la mantenibilidad del código, ya sea por nombres de variabilidad y complejidad de funciones.
- *Hotspots* de seguridad
 - 0 detectados, reflejando una buena práctica para el uso de credenciales y estructuras de seguridad en el desarrollo.

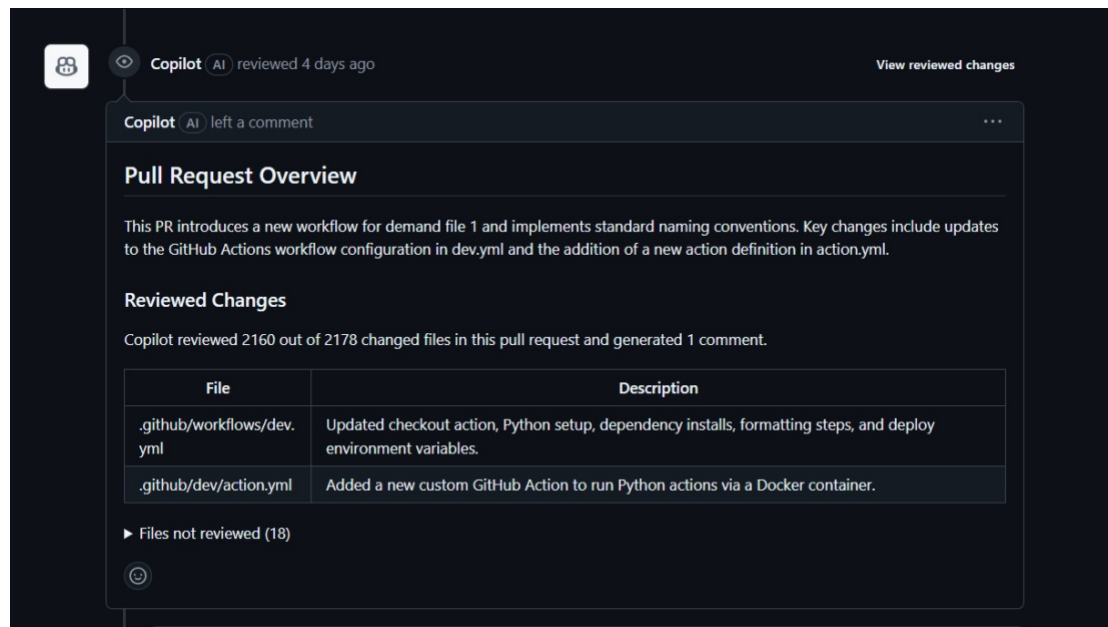
5.3.3.3. Code Review

5.3.3.3.1. Code Review mediante Copilot

Para la prueba de concepto se integra un flujo basado en *Pull Request* mediante GitHub, en el cual todo cambio enviado a ramas como *develop* o *main* debe pasar por revisión y validación antes de ser aceptado.

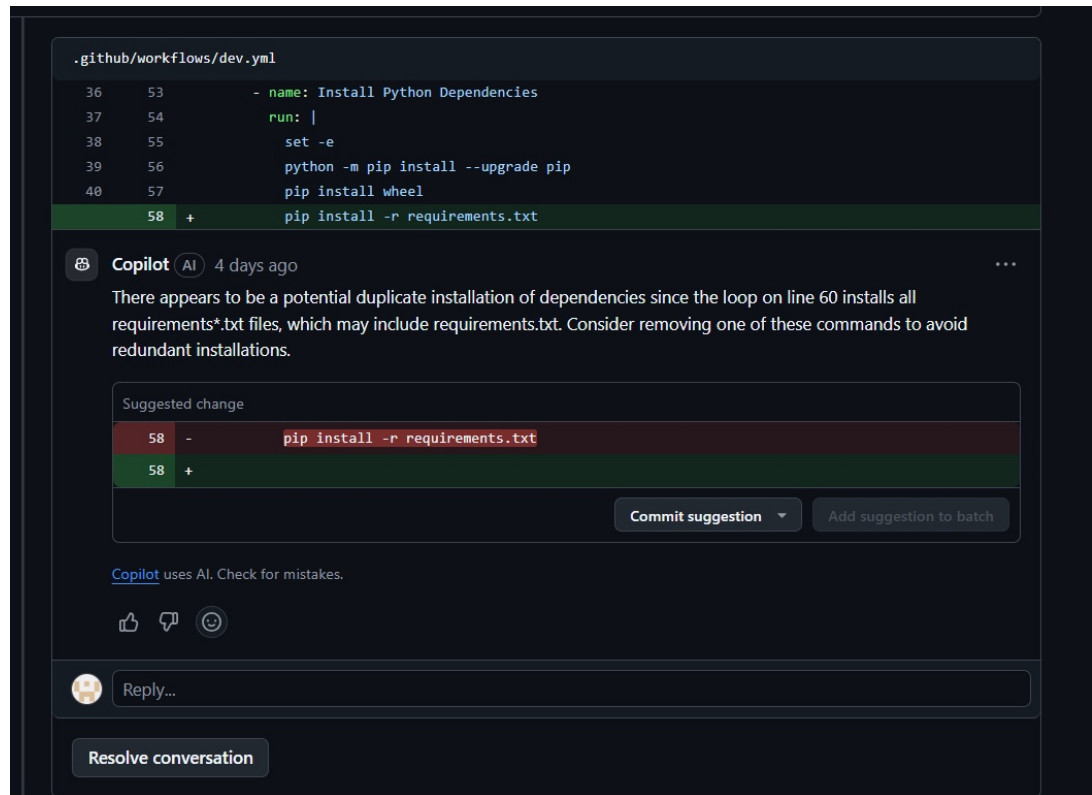
Como asistente de revisión automatizada, se utiliza GitHub Copilot, lo cual demuestra una herramienta de Inteligencia Artificial para fortalecer el cumplimiento de métricas por parte del equipo.

Figura 23: Code Review mediante Copilot 1



Fuente: elaboración propia (2025).

Figura 24: Code Review mediante Copilot 2



Fuente: elaboración propia (2025).

Durante la realización de *code review*, Copilot implementa una revisión automática del código enviado. Según se visualiza en las imágenes correspondientes al *Code Review*, se puede observar el resumen generado de la revisión de la integración que se desea realizar:

- Se detectan 2160 líneas cambiadas dentro de un total de 2178.
- Se resumen los archivos modificados entre *branches*.
 - Modificación relacionada con la instalación de dependencias y configuración del entorno.
 - Nueva acción para el desarrollo del *pipeline*.
- A su vez, Copilot genera una sugerencia concreta al detectar una instalación duplicada en las dependencias del *pipeline*. Como solución, recomienda eliminar la instrucción duplicada para evitar instalaciones repetidas.

Es necesario destacar que la incorporación de Copilot no reemplaza al revisor humano, siendo este el principal actor para la integración de soluciones en producción.

5.3.3.4. Despliegue

5.3.3.4.1. Establecimiento de parámetros del pipeline

El marco DevOps CI/CD busca parametrizar el comportamiento del *pipeline*, lo que permite automatizar tareas críticas como el *testing*, análisis de calidad y despliegue, manteniendo el control sobre los entornos de ejecución.

Para establecimiento de los parámetros clave, se configuran los siguientes componentes indispensables:

1. *Triggers* del *workflow*
2. *Branches*
3. Parámetros de entorno para el desarrollo de las acciones

Figura 25: Definición de parámetros del pipeline

```
name: RCCP INPUT AUTOMATION (Continuous Integration) DEV

on:
  pull_request: # will always use the workflow definition from master/main, not your PR!
    types: [opened, synchronize, reopened]
    branches:
      - develop
      - 'releases/**'

  push:
    branches:
      - develop

env:
  DATABRICKS_HOST: ${ secrets.DATABRICKS_HOST }
  DATABRICKS_TOKEN: ${ secrets.DATABRICKS_TOKEN }
  DATABRICKS_CLUSTER_ID: ${ secrets.DATABRICKS_CLUSTER_ID }
  DATABRICKS_BUNDLE_ENV: dev
```

Fuente: elaboración propia (2025).

5.3.3.4.2. Integración de procedimientos dentro del pipeline

Una vez que se definen los parámetros generales del *pipeline*, el siguiente paso es ejecutar las operaciones necesarias para el desarrollo del proyecto, esto incluye los siguientes pasos:

1. Instalación de dependencias
2. Configuración del entorno virtual
3. Preparación del proyecto para análisis, *testing* y despliegue.

A continuación, se desarrollan dichas etapas para la prueba de concepto mediante los *workflows* de GitHub Actions:

5.3.3.4.2.1. Instalación de dependencias y entorno virtual

Figura 26: Instalación de dependencias y entorno virtual

```
jobs:
  checks:
    runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v4
      with:
        fetch-depth: 0
    - name: Set up Python 3.9
      uses: actions/setup-python@v4
      with:
        python-version: 3.9
    - name: Set PYTHONPATH
      run: echo "PYTHONPATH=$PYTHONPATH:$(pwd)/src" >> $GITHUB_ENV
    - name: Install Python Dependencies
      run: |
        set -e
        python -m pip install --upgrade pip
        pip install wheel
        pip install -r requirements.txt
        pip install -r requirements-devel.txt
        for f in ./requirements*.txt; do pip install -r $f; done
    # Now install the Python package itself
    - name: Install This Python Package
      run: pip install -e '[devel]'
    # Check black
    - name: Check black formatting
      id: python-black-formatting
      run: |
        black ./src && echo success=true >> $GITHUB_OUTPUT
      continue-on-error: true
```

Fuente: elaboración propia (2025).

5.3.3.4.2.2. Pruebas unitarias y generación de reporte de cobertura

Figura 27: Ejecución de pruebas unitarias y generación de reporte de cobertura

```
- name: Run all tests with coverage
  id: run-pytest-cov
  run: |
    echo "Launching Unit Test"
    pytest test --cov --cov-report=xml
    continue-on-error: true

# Post Coverage for SonarQube
- name: Post Coverage Report
  uses: ./dev
  if: |
    startsWith(github.event_name, 'pull_request')
  with:
    check: post-coverage-report
    args: '--token ${ secrets.GITHUB_TOKEN }'
    continue-on-error: true
```

Fuente: elaboración propia (2025).

5.3.3.4.2.3. Ejecución de SonarQube

Figura 28: Ejecución de SonarQube

```
# Running SonarQube
- name: Run Sonarqube
  uses: sonarsource/sonarqube-scan-action@master
  with:
    args: >
      -Dsonar.python.coverage.reportPaths=coverage.xml
      -Dsonar.sources=src/
      -Dsonar.tests=test/
      -Dsonar.verbose=true
  env:
    SONAR_TOKEN: ${ secrets.SONAR_TOKEN }
    SONAR_HOST_URL: ${ secrets.SONAR_HOST_URL }
```

Fuente: elaboración propia (2025).

5.3.3.4.3. Despliegue y ejecución de solución mediante Databricks Asset Bundles

Figura 29: Despliegue y ejecución mediante Databricks Asset Bundles

```
deploy:
  name: "Deploy Bundle"
  runs-on: ubuntu-latest

  needs:
    - checks

  steps:
    - uses: actions/checkout@v4
    - uses: databricks/setup-cli@main

    - run: databricks bundle deploy
      working-directory: .
      env:
        DATABRICKS_TOKEN: ${ secrets.DATABRICKS_TOKEN }
        DATABRICKS_BUNDLE_ENV: dev

# Validate deploy and then run the bundle
pipeline_update:
  name: "Run pipeline update"
  runs-on: ubuntu-latest

  needs:
    - deploy

  steps:
    - uses: actions/checkout@v4
    - uses: databricks/setup-cli@main

    - run: databricks bundle run rccp_input_automation_job --refresh-all
      working-directory: .
      env:
        DATABRICKS_TOKEN: ${ secrets.DATABRICKS_TOKEN }
        DATABRICKS_BUNDLE_ENV: dev
```

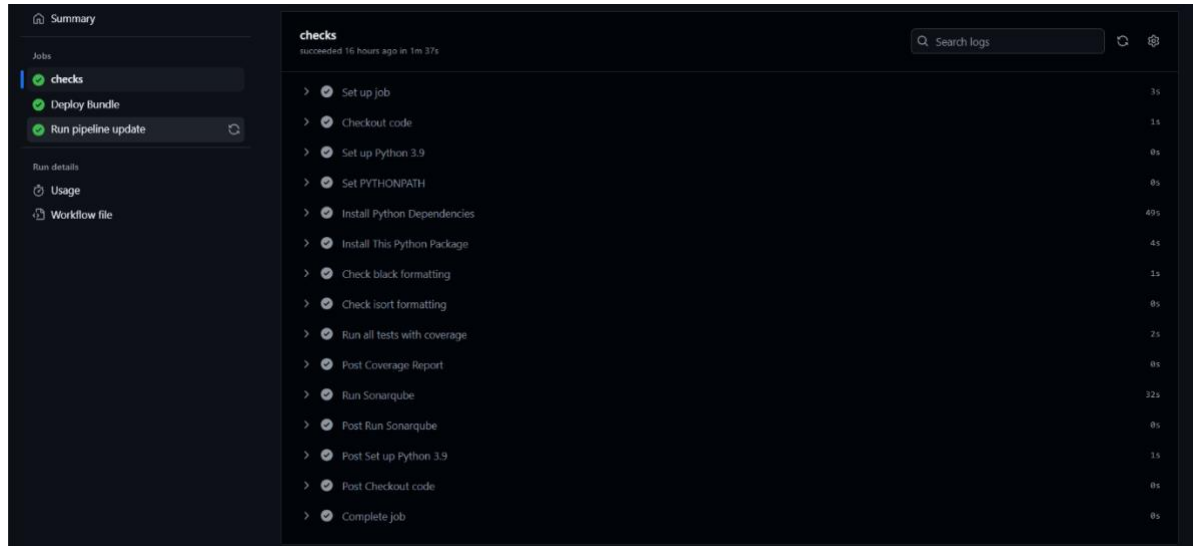
Fuente: elaboración propia (2025).

5.3.3.5. Monitoreo

La fase de monitoreo es una función esencial en el marco de DevOps CI/CD, ya que permite observar en tiempo real los procesos, ejecuciones y rendimientos de cada etapa del flujo de trabajo que se visualiza en el *pipeline* de desarrollo.

5.3.3.5.1. Visualización de los pasos para la ejecución del pipeline

Figura 30: Visualización de los pasos de ejecución del pipeline



Fuente: elaboración propia (2025).

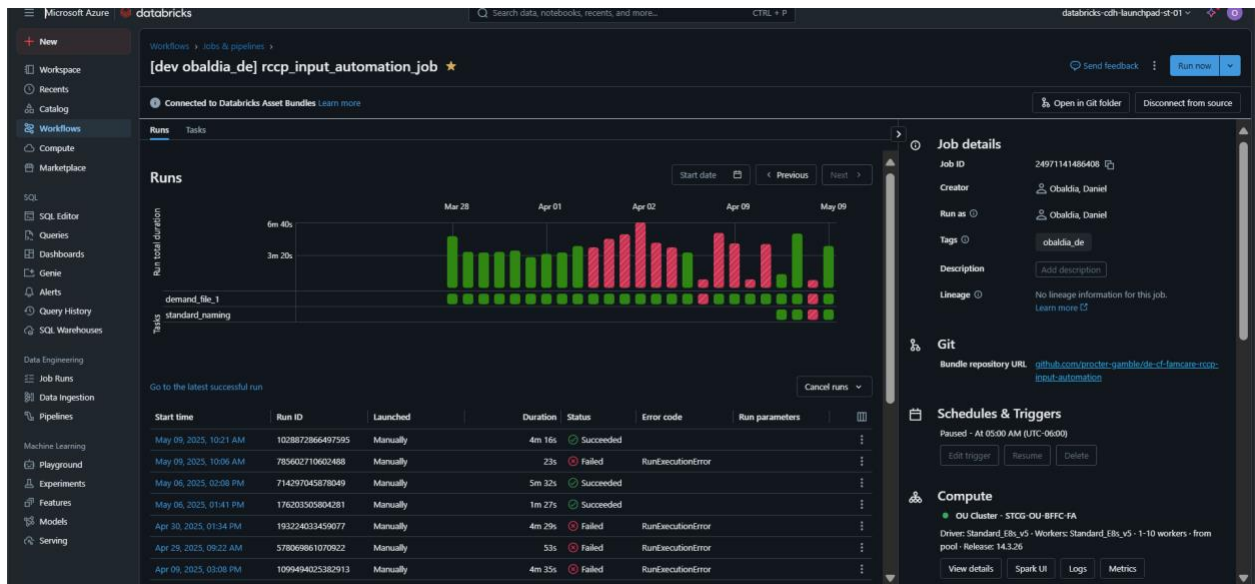
En primera instancia, es posible visualizar todo el *workflow* del *pipeline*, el cual incluye todos los pasos descritos en la fase de despliegue. Sin embargo, se debe recalcar que, si existe cualquier procedimiento erróneo, esta es la sección donde es posible verificar errores, *logs* en ejecución e indicadores de éxito. Esto permite al equipo detectar fácilmente errores en el desarrollo de funcionalidad, procesos o el *pipeline*.

5.3.3.5.2. Workflow de desarrollo en Databricks

En la figura 30, se ilustra la ejecución histórica del flujo desplegado en desarrollo, donde se observan las siguientes métricas a destacar:

- Línea de tiempo con ejecuciones exitosas o erróneas.
- Errores y excepciones recibidas, facilitando su diagnóstico.
- Posibilidad de explorar *logs* detallados por cada ejecución.
- Validación de ejecuciones manuales o automáticas.
- Métricas de duración, errores y evolución del rendimiento.

Figura 31: Workflow de desarrollo en Databricks



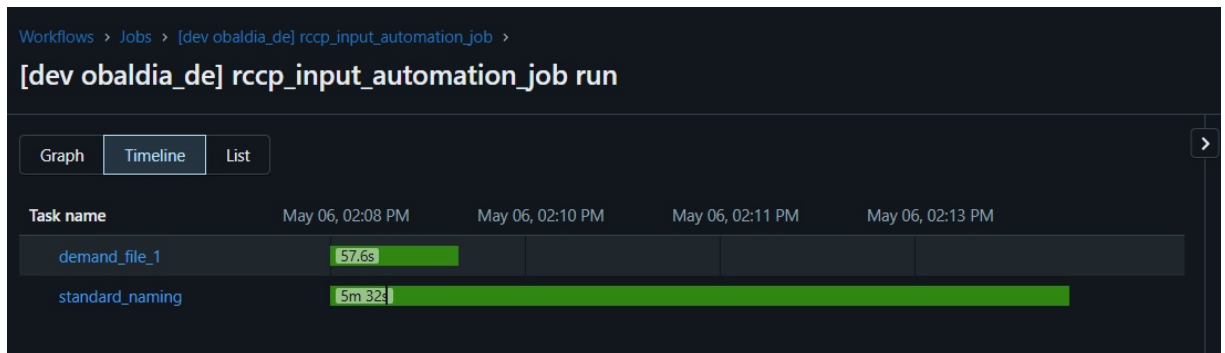
Fuente: elaboración propia (2025).

5.3.3.5.3. Medición de métricas de ejecución

Databricks proporciona métricas precisas sobre el tiempo de ejecución de cada *task* de la solución desplegada. La figura 31 presenta las dos tareas ejecutadas:

- demand_file_1: 57.6 segundos de ejecución.
- Standard_naming: 5 minutos y 32 segundos.

Figura 32: Métricas de ejecución de los tasks del pipeline



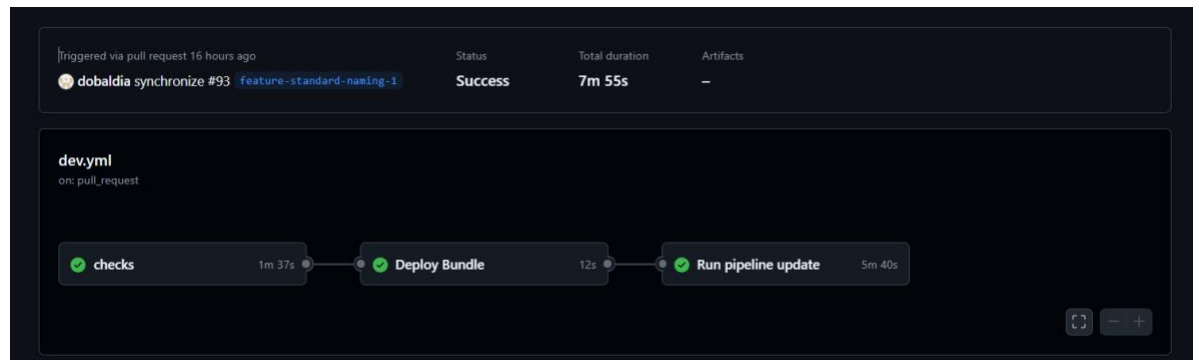
Fuente: elaboración propia (2025).

Estos datos permiten realizar el análisis de eficiencia del *pipeline*, detectar cuellos de botella y estimar tiempos de entrega más realistas para la planificación y definición del *Lead Time*.

5.3.3.5.4. Workflow del pipeline

Finalmente, GitHub Action presenta el flujo del *pipeline* indicando cómo se desarrollan las tareas según la especificación del *pipeline*.

Figura 33: Workflow del pipeline en Databricks



Fuente: elaboración propia (2025).

5.3.4. Resultados de la prueba de concepto

En esta sección, se valoran las métricas necesarias para la comparación efectiva de la prueba de concepto, valorando los procesos anteriores con el uso de *notebooks* de Databricks, al procedimiento técnico propuesto del marco de trabajo.

5.3.4.1. Especificación de métricas

5.3.4.1.1. Métrica de tiempo de ejecución de *Jobs* mediante código de Python

Este código busca conocer cuánto tarda el código en ejecutarse, tomando en cuenta todo el contenido del *notebook/job* en cuestión, los cuales procesan los mismos volúmenes de datos de la prueba de concepto.

Figura 34: Métrica de tiempo de ejecución mediante código

```
%python
from datetime import datetime

start_time = datetime.now()

# your code ...

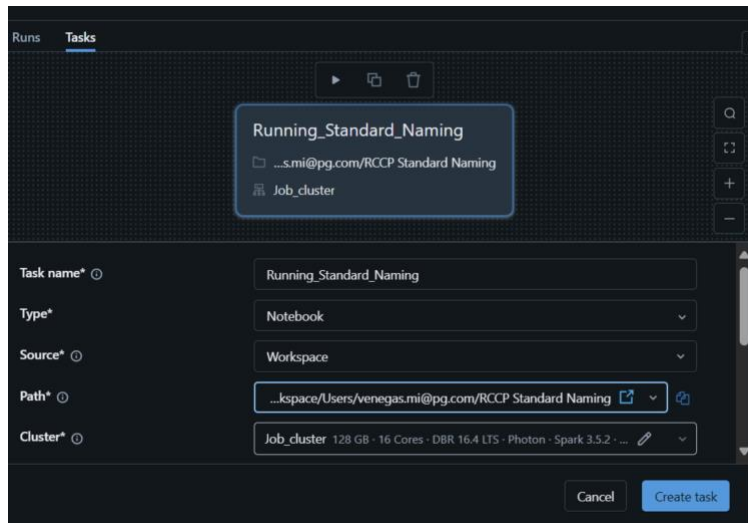
end_time = datetime.now()
duration = end_time - start_time
print(f"Notebook run duration: {duration}")
```

Fuente: elaboración propia (2025).

5.3.4.1.2. Métrica de tiempo de ejecución mediante la creación de jobs

Esta métrica crea un *job*, el cual es una tarea automatizada que ejecuta *notebooks*, *scripts* o flujos de trabajo en un clúster. En la siguiente figura, se configura la tarea, el *notebook* que se desea ejecutar y el clúster donde se llevará a cabo dicha tarea.

Figura 35: Métrica de tiempo de ejecución mediante la creación de jobs



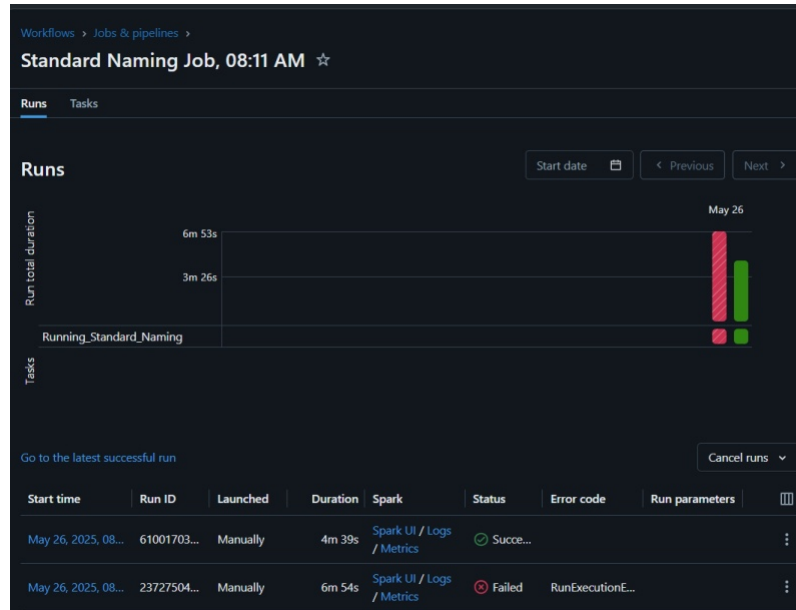
Fuente: elaboración propia (2025).

5.3.4.2. Resultados de las métricas antes y después de la ejecución de la prueba

A continuación, se presentan dos ejecuciones con las métricas especificadas anteriormente, con el objetivo de validar el tiempo de ejecución del nuevo marco estandarizado de DevOps CI/CD. Esto a partir de las tareas demostradas en la prueba de concepto.

5.3.4.2.1. Ejecución de la tarea Standard Naming con el procedimiento anterior

Figura 36: Ejecución de la tarea Standard Naming con el procedimiento anterior



Fuente: elaboración propia (2025).

Figura 37: Ejecución de la tarea Standard Naming con el procedimiento anterior mediante código

```
%python
end_time = datetime.now()
duration = end_time - start_time
print(f"Notebook run duration: {duration}")
```

Notebook run duration: 0:02:16.214930

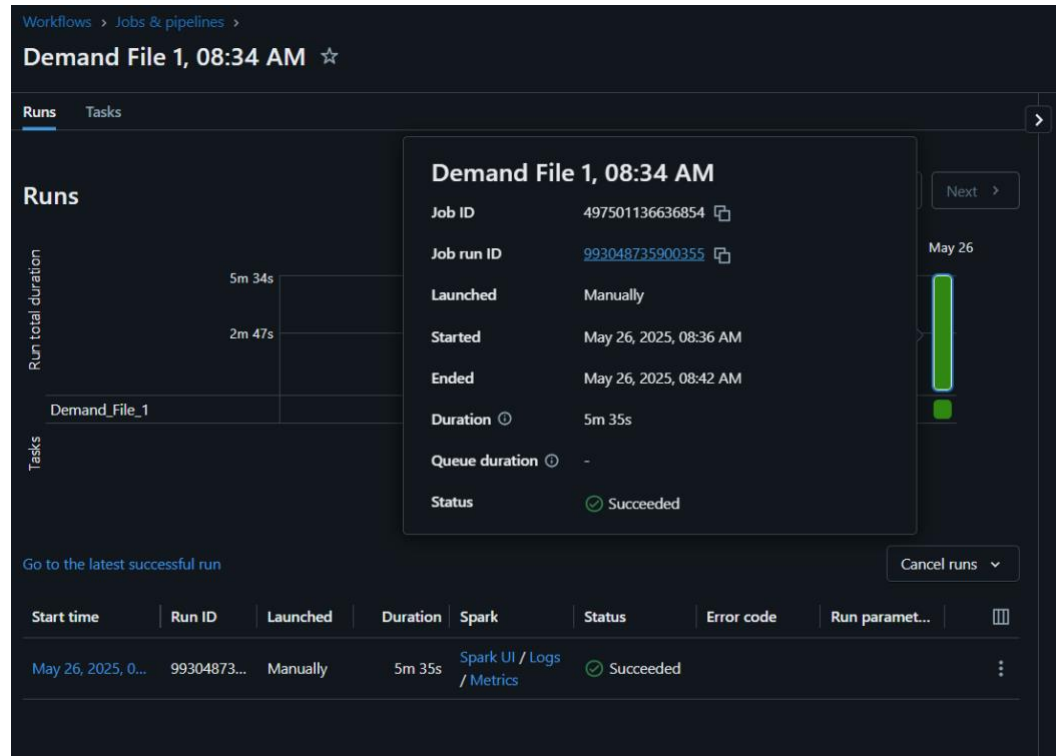
Fuente: elaboración propia (2025).

Tomando en cuenta la primera ejecución, se visualiza un tiempo de 6 minutos y 54 segundos en la primera ejecución. En la segunda ejecución, se presenta un tiempo de 4 minutos y 39 segundos de ejecución. Esto mediante el método de creación y ejecución de *jobs* en Databricks.

Visualizando la tercera ejecución correspondiente a código en Python, el tiempo de ejecución fue de 2 minutos y 16 segundos.

5.3.4.2.2. Ejecución de la tarea Demand File 1 con el procedimiento anterior

Figura 38: Ejecución de la tarea Demand File 1 mediante la creación de jobs



Fuente: elaboración propia (2025).

Figura 39: Ejecución de la tarea Demand File 1 mediante código

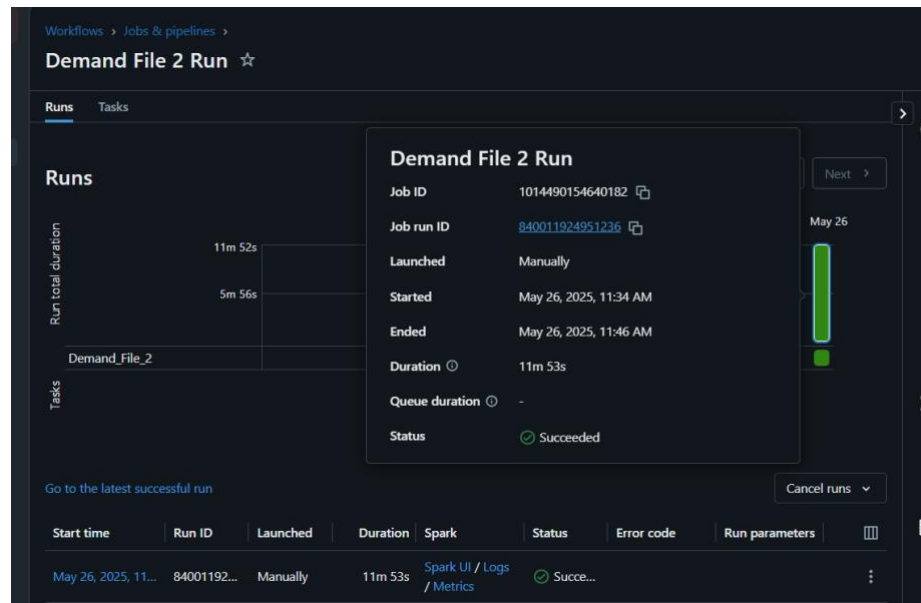


Fuente: elaboración propia (2025).

Explorando los resultados obtenidos, en el primer caso se puede verificar que manualmente el *job* Demand File 1 dura 5 minutos y 35 segundos en ejecutarse cuando se implementa la creación de la tarea mediante Databricks. En el segundo caso, puede visualizarse un tiempo de 3 minutos y 5 segundos para la ejecución de todas las funcionalidades del *notebook*, utilizando el método del código de Python.

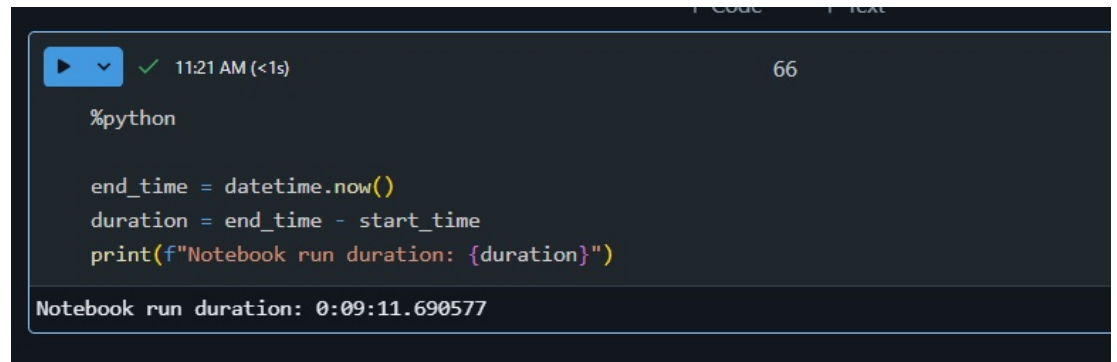
5.3.4.2.3. Ejecución de la tarea Demand File 2 con el procedimiento anterior

Figura 40: Ejecución de la tarea Demand File 2 mediante la creación de jobs



Fuente: elaboración propia (2025).

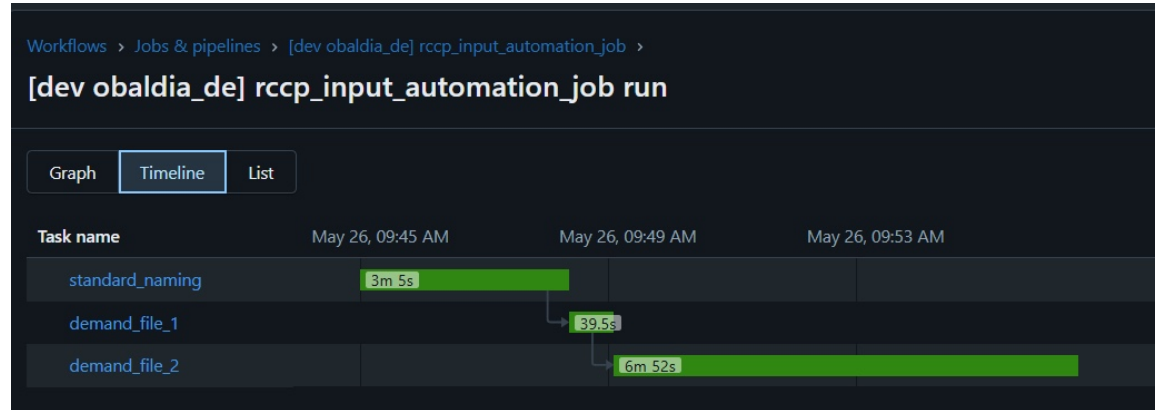
Figura 41: Ejecución de la tarea Demand File 2 mediante código



Fuente: elaboración propia (2025).

Explorando los resultados obtenidos, en el primer caso se puede visualizar que el tiempo de ejecución de *demand_file 2* es de 11 minutos y 53 segundos, esto mediante la creación de la tarea utilizando el *job* de Databricks. En el segundo caso utilizando código de Python, se puede visualizar un tiempo de ejecución de 9 minutos y 11 segundos.

5.3.4.2.4. Ejecución de la tarea mediante el nuevo procedimiento



Fuente: elaboración propia (2025).

1. Standard Naming Task

Utilizando *pipeline* automatizado de desarrollo, es posible visualizar que el tiempo de ejecución de la tarea de *standard_naming* es de 3 minutos y 5 segundos.

2. Demand File 1

Mediante el *pipeline* automatizado de desarrollo, se puede verificar que el tiempo de ejecución de la tarea *demand_file_1* es de 39 segundos.

3. Demand File 2

Con la ejecución del *pipeline* automatizado, es posible comprobar que el tiempo de ejecución para la tarea de *demand_file_2* es de 6 minutos y 52 segundos.

5.3.4.2.5. Resultados de ejecuciones

Tabla 21: Resultados de las ejecuciones del pipeline

Tarea realizada	Procedimiento original	Procedimiento con DevOps CI/CD	Beneficio esperado
Standard Naming	4:01 min (promedio entre 6:54, 4:39 y 2:16)	3: 05 min	Reducción de 56 segundos (23.32%)
Demand File 1	4:20 min (promedio entre 5:35 y 3:05)	0:39 min	Reducción de 3:41 min (85%)
Demand File 2	10:32 min (promedio entre 11:53 y 9:11 min)	6:52 min	Reducción de 3:40 min (34.81%)

Fuente: elaboración propia (2025).

Los resultados obtenidos a partir de las tres tareas permiten evidenciar una mejora sustancial en el rendimiento operativo gracias al uso del marco DevOps CI/CD. Estos resultados demuestran eficiencia técnica, validando su aplicabilidad como estándar para futuras tareas dentro del equipo.

5.3.4.2.6. Resultados de ejecuciones según métricas DevOps

Tabla 22: Resultados de las ejecuciones según métricas DevOps

Métrica	Ejecución previa	Aplicación con el marco DevOps CI/CD
Lead Time	No es correcto valorar el <i>Lead Time</i> , ya que no existen despliegues continuos.	Es correcto afirmar que el <i>Lead Time</i> corresponde al momento en que el requerimiento fue creado hasta que el código es desplegado, por lo que se puede suponer que, si un requerimiento fue inicializado a las 8:00 a.m., se desarrolla y posteriormente se ejecuta en el <i>pipeline</i> . Si se valora todo el <i>pipeline</i> y los ejemplos de la prueba de concepto, se puede confirmar que el <i>Lead Time</i> empieza desde que se asigna el requerimiento, se desarrolló y los 10 minutos con 36 segundos del despliegue del <i>pipeline</i> .

Métrica	Ejecución previa	Aplicación con el marco DevOps CI/CD
Deployment Frequency.	Un despliegue semanal sin visualización en desarrollo o producción.	Despliegue por cada <i>feature</i> desarrollado, con integración en desarrollo o producción.
Change Fail Rate / Time to restore	Se desarrolla en Databricks y hasta que se ejecuta de nuevo el <i>notebook</i> se maneja el problema.	El <i>pipeline</i> automatizado permite conocer cuando existe un problema mediante la ejecución de este, notificando al usuario en cuestión. Esto se verifica mediante la programación que se configure en el <i>pipeline</i> .
Tiempo del despliegue	El total de esta ejecución es de 18 minutos con 53 únicamente de las tareas del <i>pipeline</i> .	La ejecución abarca todos los pasos del <i>pipeline</i> con una duración de 10 minutos y 36 segundos.

Fuente: elaboración propia (2025).

5.4. Análisis costo-beneficio

Analizando la situación actual, estos corresponden a los precios que ofrecen las plataformas a la organización, tomando en cuenta al usuario y uso por mes. Se toman en cuenta las herramientas necesarias para el desarrollo de *pipeline* dentro del marco de DevOps, siendo utilidades de planificación, desarrollo, *testing* y, por último, despliegue. A su vez, se cuenta con la aproximación estimada del uso de la plataforma Databricks para su gestión y exploración de los datos del equipo.

La obtención de estos montos en dólares corresponden a licencias que actualmente cuenta la organización para la adquisición de estas tecnologías o herramientas, siendo documentación interna específicamente para el área de *Data & Analytics* de la organización.

A continuación, se presentan los costos por usuario por el uso de las herramientas necesarias para el flujo del marco propuesto:

5.4.1. Determinación de los costos totales proyecto

Tabla 23: Determinación de los costos totales del proyecto

Plataforma	Descripción	Costo en dólares
GitHub	Por usuario al año	\$240
Confluence	Por usuario al mes	\$3.99
Jira Software	Por usuario al mes	\$14.97
SonarQube	Por la gestión de 100.000 líneas de código	\$660
Databricks	<u>Clúster mediante el servicio del <i>Launchpad</i> para la unidad de cuidado familiar.</u> <ul style="list-style-type: none"> Actualmente, el uso del clúster incluye un largo número de trabajadores, no solo el equipo dedicado de AI Engineering. Esto es una aproximación del costo del clúster actual para la Unidad de Cuidado Familiar. 	\$10.000
	<u>Utilización de un clúster dedicado para las necesidades</u> <ul style="list-style-type: none"> El cálculo se estima para los 7 trabajadores actuales del equipo al mes. Esta solución busca cubrir las necesidades del equipo incluyendo soporte a DAB, entornos separados de desarrollo y producción, acceso compartido a <i>notebooks</i>, <i>sql</i>, <i>spark</i>, clúster escalable y auto-terminable. 	\$2301

Fuente: elaboración propia (2025).

5.4.2. Cálculo del retorno de inversión

En esta sección, se calcula el retorno sobre la inversión (ROI) de la propuesta del marco de estandarización, la cual toma en cuenta las necesidades del equipo de Ingeniería de Inteligencia Artificial, desde sus especificaciones técnicas y operativas hasta sus necesidades en los métodos y plataformas de trabajo. Para el análisis de la viabilidad de la propuesta, se abordan dos enfoques complementarios de ROI:

- a. **ROI Clásico (financiero):** comparativa de costos entre la situación actual y la propuesta, para estimar el ahorro económico y verificar si el costo propuesto es suficiente para las necesidades del equipo.
- b. **ROI Estratégico (beneficios organizacionales):** análisis del valor generado por mejoras de productividad, calidad y alineación con los estándares corporativos DevOps CI/CD y el AI Readiness Criteria.

5.4.2.1. ROI Clásico: análisis de costos de la solución propuesta para la plataforma Databricks

El ROI clásico se basa en la comparación del costo de la plataforma actual versus el costo de la solución propuesta, y calcula el retorno en función del ahorro obtenido. La fórmula empleada es:

$$ROI = \left(\frac{\text{Costo Actual} - \text{Costo propuesto}}{\text{Costo propuesto}} \right) \times 100$$

$$ROI = (\text{Ahorro Neto} \div \text{Costo de la Inversión}) \times 100\%$$

Tabla 24: Costos de la nueva solución de Databricks

Costo relacionado	Solución actual de Databricks	Solución propuesta
Costo mensual estimado	\$10.000	\$2.300
Costo anual estimado	\$120.000	\$27.600
Ahorro en costos	N/A	\$92.400 (reducción 77%)

Fuente: elaboración propia (2025).

En este caso, la inversión es el costo del nuevo clúster dedicado a las funciones necesarias para el equipo de Inteligencia Artificial, tomando en cuenta sus necesidades técnicas y operativas. En perspectiva estos montos radican en lo que actualmente

invierte la Unidad de Cuidado familiar en comparación a la solución propuesta de la página de *Databricks*.

Como se observa, la propuesta reduce drásticamente el gasto operativo. El costo anual bajaría de \$120.000 a \$27.600, ahorrando aproximadamente \$92.000 al año, lo que equivale a una reducción del 77% en los gastos de la plataforma. En términos de ROI, esto se ve ejemplificado de la siguiente manera:

$$ROI = \left(\frac{\$120.000 - \$27.600}{\$27.600} \right) \times 100 = 334,78\%$$

Desde la perspectiva financiera, utilizar otro clúster adaptado al equipo es altamente rentable, sin embargo, hay que destacar que esto solo toma en cuenta al equipo de Ingeniería de Inteligencia Artificial encargada de los proyectos de manufactura de la Unidad de Cuidado Familiar.

5.4.2.2. ROI Financiero: solución propuesta mediante las especificaciones técnicas y operativas

Tabla 25: Costos de las especificaciones técnicas

Plataforma	Costo mensual o anual	Costo total anual
GitHub	\$240 /usuario / año x 7 usuarios	\$1680
Confluence	\$3.99 /mes /usuario x12 meses x 7 usuarios.	\$344.44
Jira	\$14.97/mes/ usuario x12 meses x 7 usuarios	\$1.256.28
SonarQube	Gestión de 100.000 líneas de código	\$660
Databricks	Clúster específico \$2.301 / mes x 12 meses	\$27.612
Total		\$31.541.72

Fuente: elaboración propia (2025).

Con base en estos costos, se calcula el beneficio financiero, el cual es la diferencia de costo frente al modelo actual, que es el siguiente:

$$Beneficio\ anual = \$120.000 - \$31.542.72 = \$88.457.28$$

Seguidamente, es posible calcular el ROI financiero aplicando la fórmula especificada previamente:

$$ROI = \left(\frac{\$88.457.28}{\$31.542.72} \right) \times 100 = 280.4\%$$

Según el retorno de inversión financiero, por cada dólar invertido en la nueva solución, se recuperan \$2.80 dólares en términos técnicos-operativos.

5.4.3. Cálculo del ahorro en tiempo de ejecución

Como parte del análisis costo-beneficio asociado a la propuesta del marco DevOps CI/CD, se realizó una estimación del valor económico generado únicamente por el ahorro de tiempo de ejecución de las tareas del *pipeline*. Esta estimación se sustenta en los resultados de la prueba de concepto, donde se comparan los tiempos operativos del proceso actual y la integración del *pipeline* automatizado.

1. Ahorro semanal por persona (cada vez que debe hacer un despliegue)

$$9 \text{ min} \times 5 \text{ días} = 45 \text{ minutos}$$

2. Jornada laboral semanal

$$\frac{45}{2400} \text{ min} \times 100 = 1.875\%$$

3. Beneficio económico mensual estimado

- Salario mensual promedio de un *Data Engineer* es de \$2900. (Jobicy, 2025).
- Ahorro mensual por persona

$$\$2900 \times 1.875\% = \$54.375$$

- Ahorro mensual total para todo el equipo

$$\$54.375 \times 7 = \$372.87 \text{ USD}$$

5.4.4. Necesidades técnicas del equipo

Es necesario corroborar con los requerimientos del equipo de Ingeniería de Inteligencia Artificial. A continuación, se justifican las capacidades de la nueva solución según los objetivos del proyecto, tales como DevOps CI/CD, calidad, DAB, ambientes de desarrollo y producción, entre las demás necesidades vistas se encuentran:

5.4.4.1. Integración CI/CD y despliegue continuo

El clúster propuesto permite la implementación de *pipelines* de CI/CD para los proyectos del equipo. Databricks, a su vez, soporta la integración con *git* para el control de versiones, así como la integración de herramientas de CI/CD, que, en su defecto, se utilizaría GitHub Actions.

El equipo, por su parte, podrá construir, testear y desplegar código de forma automatizada y frecuente, cumpliendo con los flujos de DevOps. A su vez, la disponibilidad de un entorno dedicado evita las demoras que existen en el Launchpad para ejecutar jobs o pruebas de desarrollo, habilitando despliegues más rápidos y confiables.

5.4.4.2. Estándares de calidad y buenas prácticas

El equipo puede incorporar controles de calidad en su proceso, esto incluye el uso de repositorios Git, *workflow* de CI/CD, revisión de la calidad del código, *code review*, *unit testing* y monitoreo de las funcionalidades después de cada despliegue.

La obtención de un clúster propio garantiza consistencia en los entornos, de modo que lo validado en desarrollo será integrado correctamente en producción. Además, se puede aprovechar la funcionalidad de Databricks para mantener trazabilidad de cambios de código y *rollback* si es necesario.

5.4.4.3. Databricks Asset Bundles (DAB)

La propuesta contempla el soporte mediante Databricks Asset Bundles, una herramienta con un enfoque en estandarizar las entregas del equipo de Ingeniería de Inteligencia Artificial. DAB permite versionar y testear código, modelos y *pipelines*, facilitando la gestión de múltiples ambientes y la adopción de prácticas DevOps CI/CD.

Al contar con un clúster dedicado y compatible con el despliegue de versiones mediante Databricks Asset Bundles, es posible asegurar que cada integración de código cumple con los criterios de calidad y estructura requeridos. De esta forma, se cierra la brecha existente entre el Launchpad y el equipo para soluciones efectivas y ambientes de desarrollo y producción.

5.4.4.4. Separación de entornos (*dev/prod*)

Con la propuesta del marco de trabajo, es posible configurar una clara separación entre el entorno de desarrollo y producción. Por ejemplo, es posible tener un clúster o configuración para desarrollo y pruebas, y otro para la realización de pruebas, que distingan el trabajo realizado de cada AI Engineer. Esto garantiza la estabilidad de soluciones e integración de la metodología DevOps CI/CD.

5.4.5. Beneficios organizacionales y de productividad

La propuesta de estandarización de un marco de trabajo mediante DevOps CI/CD incorpora mejoras de eficiencia y calidad que impactan directamente a la organización. Según *Accelerate*: “Las organizaciones con equipos de alto desempeño de *TI* deben ser consistente en los resultados de negocio, tanto de rentabilidad, mercado y productividad”.

5.4.5.1. Reducción de Lead Time

Con la integración de CI/CD y la eliminación de cuellos de botella del Launchpad, el tiempo medio para la integración de código a producción disminuye, debido a que se propone que las integraciones sean progresivas, no únicamente cuando el proyecto se encuentra en etapas avanzadas.

Actualmente, procesos manuales pueden significar semanas para probar e implementar un nuevo *feature*, con el proceso de automatización y generación de valor mediante despliegues continuos, esto puede reducirse de días a horas. Esta disminución del *Lead Time* permite la entrega de valor constante al negocio. Por ejemplo, si se calcula que para un proyecto la entrega de resultados es cada 2 semanas, bajo el nuevo marco puede lograrse en 2 días y verse implicado en un *Branch* de desarrollo, únicamente esperando el *code review* para el despliegue en producción.

5.4.5.2. Incremento en la frecuencia de despliegues

Al implicar *pipelines* automatizados, es posible permitir al equipo realizar despliegues continuos a producción con una mayor frecuencia. En lugar de esperar a una acumulación de múltiples cambios, es esperable que el tiempo de despliegue sea en pequeños lotes o, por ende, por *feature*. Esto permite que los procesos se agilicen, gestión de errores más frecuente, *code review* frecuente y entrega de valor al negocio de forma progresiva.

Si se supone que el equipo desplegaba una solución cada mes, el nuevo flujo permite realizarlo semanalmente o en casos urgentes, diariamente. En general, más lanzamientos permiten un valor más constante. Para evaluarlo dentro del ROI, se toma en cuenta la utilización de los desarrolladores, donde no se encuentran bloqueados en un *release*, sino que entregan frecuentemente soluciones.

5.4.5.3. Menor tasa de fallos en producción (Change Failure Rate)

La calidad mejorada del proceso tomando en cuenta pruebas automáticas, controles de versiones y diferenciación ambientes reduce la proporción de despliegues que generan los errores o incidentes en producción. Por ejemplo, si hoy el 20% de los cambios causan algún error crítico, se podría aspirar a que ese número sea reducido a 10%, mediante el uso de SonarQube, *unit testing* y la implicación de un ambiente de desarrollo diferenciado del de producción, implicando en cambios únicamente validados previamente.

Es claro recalcar que DevOps es una fuente directa de alto desempeño, donde es posible verificar con porcentajes la reducción en el trabajo no planificado y correcciones de estos, de esta forma, se invierte en la creación de nuevas funcionalidades o mejoras de automatización. En el contexto de este proyecto, se refleja en horas en las que los ingenieros deben consumir en apagar incendios o volver a procesar funcionalidades creadas.

5.4.5.4. Calidad del código y colaboración

Al seguir las pautas del marco DevOps CI/CD utilizando repositorios, seguimiento de cambios, documentación automatizada, revisión de estructura de código, entre los demás beneficios del marco, es posible mejorar el rendimiento y visibilidad del ciclo de vida de proyectos del equipo de Ingeniería de Inteligencia Artificial. En términos de beneficio, es difícil de medir, pero tomando en cuenta estos beneficios, se puede desarrollar una mejora organizativa, que implica mejor utilización de recursos y tiempo, lo cual contribuye indirectamente al ROI, al mantener el equipo enfocado en procedimientos ágiles, integrando valor constante.

5.4.5.5. Reducción del tiempo improductivo del Launchpad

Uno de los impactos financieros más notorios es el hecho de un clúster dedicado al equipo de Ingeniería de Inteligencia Artificial. El Launchpad, al ser multiusuario y tener recursos limitados, es probable que presente problemas al ejecutar cargas intensivas o problemas de almacenamiento al realizar procedimientos integrados en los proyectos del equipo. Con la incorporación de un clúster propio que cumpla con los lineamientos del marco propuesto, es posible que los recursos siempre estén disponibles para el equipo, resultando en la disminución de tiempos muertos y sobre todo la adaptación a las métricas de AI Readiness Criteria.

Suponiendo que los ingenieros presentan problemas porque se perdían dos horas por reinicios del entorno, equivalentes a un costo de \$50/hora, con la propuesta del marco, es posible aumentar significativamente la capacidad productiva del personal en tiempos de desarrollo, buscando la disponibilidad operacional mediante un clúster que cumpla con las necesidades del entorno ágil que busca la propuesta de DevOps CI/CD.

5.4.5.6. Cumplimiento de criterios de AI Readiness y estándares corporativos

Con la propuesta de la metodología DevOps CI/CD, es posible incorporar los estándares del AI Readiness Criteria, mitigando riesgos de incumplimiento de seguridad y calidad, lo que facilita la integración de las soluciones informáticas del equipo, logrando proyectos escalables que brindan un beneficio financiero mayor.

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

En términos de ROI, el cumplimiento con el AI Readiness Criteria es visto como la evasión de costos potenciales, implicando desde el retrabajo de un proyecto o la falta de cumplimiento con requisitos o estándares organizacionales.

6. Conclusiones

El objetivo de esta sección es demostrar los hallazgos obtenidos en el proyecto y que se cumplieron los entregables definidos en cada etapa de este. Para ello, se presentan las conclusiones derivadas de los descubrimientos según cada objetivo, haciendo referencia a la sección del Análisis de Resultados.

Las conclusiones abordan desde el diagnóstico del estado actual del equipo de Ingeniería de Inteligencia Artificial hasta la implementación y validación de un marco de trabajo basado en la metodología DevOps. A través de este enfoque, se logró estructurar un flujo de trabajo estandarizado, automatizado y alineado con las métricas de la empresa, permitiendo mejorar en la calidad técnica de los entregables y la eficiencia de los procesos.

A continuación, se detallan los hallazgos según cada uno de los objetivos específicos del proyecto:

6.1. Conclusiones del objetivo 1

Con base en los resultados del primer objetivo específico correspondiente a: “Analizar las necesidades y limitaciones actuales del equipo de Inteligencia Artificial, en relación al marco de trabajo actual, para identificar los requerimientos técnicos y operativos necesarios para implementar correctamente la metodología DevOps CI/CD dentro de los proyectos de la Unidad de Cuidado Familiar”, se concluye:

- El 100% de los proyectos analizados iniciaron sin una planificación formal ni metodologías definidas, lo que impidió el cumplimiento del AI Readiness Criteria, como se evidencia en las secciones [4.1.1](#), [4.1.2](#) y [4.1.3](#). Esto generó inconsistencias en la calidad, escalabilidad y sostenibilidades de las soluciones desarrolladas.
- El principal cuello de botella que se identificó se encuentra en la fase de planificación y exploración de datos debida a las limitaciones en la comunicación con el negocio, como se evidencia en las secciones [4.1.1](#), [4.1.2](#) y [4.1.3](#).
- El equipo no documenta formalmente las fases del ciclo de vida de proyectos abarcando desde la planificación y el monitoreo, situación que fue validada en ambas entrevistas, como se evidencia en las secciones [4.1.1](#), [4.1.2](#) y [4.1.3](#). Esto dificulta la transferencia de conocimiento y la escalabilidad de soluciones.

- La ausencia de la integración de un *pipeline* de desarrollo, integración continua, control de versiones, verificación de calidad de código, pruebas automatizadas, así como el uso inconsistente de herramientas como Jira y Confluence, confirma que no se aplica una cultura de mejora continua o la metodología DevOps CI/CD, según se evidencia en las secciones [4.1.1](#), [4.1.2](#), [4.1.3](#) y [4.1.5](#).
- Es posible verificar que el AI Readiness Criteria es un conjunto de métricas y parámetros que evidencian cómo se puede alcanzar la calidad, seguridad y completitud de las soluciones basadas en Ingeniería de Inteligencia Artificial. Esto se visualiza en las secciones [4.1.3](#), [4.1.4](#), y [4.1.5](#).
- El procedimiento de integración con el equipo de Site Reliability Engineering carece de una estandarización para todos los proyectos, limitando la formalización de acuerdos operativos, como se evidencia en la sección [4.1.5](#).
- Es necesario aplicar los lineamientos de documentación para la integración de un proyecto con el equipo de Site Reliability Engineering, los cuales se evidencian en la sección [4.1.5](#).
- Se identifican métricas de la metodología DevOps y Lean que pueden medir la eficiencia del equipo para el despliegue de soluciones, permitiendo valorar la efectividad del marco DevOps CI/CD, el cual se evidencia en la sección [4.1.6](#). Esto permite registrar métricas, configuraciones y soluciones medibles en términos de eficiencia y productividad.
- Se evidencia que es necesario el uso de herramientas que faciliten la gestión y el despliegue de proyectos, destacando Databricks Asset Bundles y GitHub Actions. Esto se visualiza en la sección [4.1.6](#). Esto permite la integración de marcos de CI/CD aplicables a los proyectos del equipo, estandarizando el procedimiento de desarrollo y despliegue.
- Se establece que el uso de *pipelines* CI/CD requiere control de versiones, pruebas automatizadas, validación del código y despliegue continuo; prácticas que no se evidencian en la gestión actual del equipo, como se verifica en la sección [4.1.6](#).
- Se concluye que, para cumplir con el marco propuesto, el equipo debe adoptar las prácticas operativas y técnicas descritas en los marcos de referencia, desde la planificación hasta el despliegue, como se evidencia en la sección [4.1.6](#).

6.2. Conclusiones del objetivo 2

Con base en los resultados del segundo objetivo específico correspondiente a: “Diseñar el marco de trabajo basado en la metodología DevOps CI/CD, a partir de los requerimientos técnicos y operativos identificados en el análisis de la situación actual, asegurando la adopción de las mejores prácticas ágiles que sean adaptables al desarrollo de proyectos del equipo de Inteligencia Artificial.

se concluye:

- Para la resolución de la configuración de CI/CD y despliegues manuales, se define como estrategia el uso de Databricks Asset Bundles y GitHub Actions, permitiendo la automatización de integraciones, validaciones y despliegues con control de versiones, tal como se evidencia en la sección [5.1.1](#).
- Se propone integrar SonarQube dentro del *pipeline* CI/CD como herramienta de validación continua de calidad de código y la cobertura de este, tal como se muestra en la sección [5.1.1](#).
- Se establece *pytest* como *framework* de pruebas automatizadas, asegurando la ejecución de pruebas unitarias dentro del *pipeline*. Esto se visualiza en la sección [5.1.1](#).
- Se formaliza el uso de un GitFlow para habilitar la trazabilidad desde el desarrollo hasta producción, como se evidencia en la sección [5.1.1](#). Esto permite la estandarización del procedimiento de desarrollo y su integración en el control de versiones de los proyectos.
- Se establece como requerimiento técnico el proceso de Code Review, aplicándose como recurso obligatorio en fases de desarrollo y producción. Esto se visualiza en la sección [5.1.1](#). Esto permite la finalización de validación técnica implicando en soluciones más factibles y la prevención de errores.
- Para la documentación técnica, es necesario el uso de Confluence como repositorio central, alineado con el marco del equipo de SRE, permitiendo estructurar componentes, procesos, dependencias y la arquitectura del proyecto. Esto se evidencia en la sección [5.1.1](#).
- Se estandariza la documentación técnica-administrativa para los proyectos del equipo, incluyendo modelo de soporte, *stakeholders*, infraestructura, flujo de datos, acceso, arquitectura y plantillas para la transición a producción, tal como se evidencia en la sección [5.1.2](#).

- La fase de planificación del marco establece la aplicación de proceso Lean y DevOps integrando herramientas como Confluence, tableros Kanban para la gestión de requerimientos y establecimiento de métricas, evidenciándose en la sección 5.1.2.
- Se formaliza la evaluación inicial de estándares del AI Readiness Criteria desde la etapa de planificación, incluyendo seguridad, documentación, pruebas y métricas, lo cual permite identificar brechas técnicas desde etapas tempranas del ciclo de vida del proyecto, tal como se muestra en la sección 5.1.2.
- La integración a producción debe incluir documentación de estrategia de CI/CD, *secrets* de GitHub, *test cases* y dependencias, cumpliendo con el AI Readiness Criteria en sus dimensiones técnicas y operativas, como se visualiza en la sección 5.1.2.

6.3. Conclusiones del objetivo 3

Con base en los resultados del tercer objetivo específico correspondiente a: “Comparar las diferencias operacionales en términos de eficiencia, tiempos de entrega y calidad de las soluciones informáticas de la gestión de proyectos de inteligencia Artificial entre la situación actual y la propuesta de estandarización basada en DevOps CI/CD, con el propósito de validar las mejoras implementadas en el ciclo de vida de los proyectos”, se concluye:

- La recopilación de requerimientos y de documentación se estandariza a Confluence, mediante el uso de plantillas de historias de usuario, con criterios DevOps y de AI Readiness Criteria, como se evidencia en la sección 5.2. Esto implica un beneficio a los cuellos de botella identificados en la fase de planificación.
- En el marco que se propone es necesario incorporar la validación técnica, evaluando estándares de seguridad, documentación y pruebas automatizadas como parte del *pipeline* propuesto, esto se visualiza en la sección 5.2. Se desarrollan prácticas de *CI/CD* que permiten la mejora continua en la validación del pipeline automatizado.
- La estructura de desarrollo se estandariza a Databricks Asset Bundles, con una definición clara de entornos, control de versiones y automatización de proceso, tal como se muestra en la sección 5.2. Se permite estandarizar el marco de trabajo del equipo mediante el uso de una herramienta que optimiza las prácticas *CI/CD*.

- El control de versiones, anteriormente inconsistente, fue formalizado mediante el uso obligatorio de GitHub y una estrategia de Gitflow, visualizándose en la sección 5.2. Se considera una validación técnica definida que estandariza los procedimientos mínimos y necesarios para fases de desarrollo y pruebas.
- Se configura la parametrización del *pipeline* a nivel de herramientas, procedimientos y definición de *branches*, evidenciándose en la sección 5.2. Esto permite estandarizar las prácticas, herramientas y soluciones que puede desarrollar el equipo de Ingeniería de Inteligencia Artificial.
- La transición de proyectos con el equipo de SRE incluye la documentación de *onboarding* técnico, documentación del modelo de datos y procedimientos como herramientas de desarrollo, como se visualiza en la sección 5.2.

6.4. Conclusiones del objetivo 4

De acuerdo con los resultados del cuarto objetivo específico correspondiente a: “Elaborar una prueba de concepto en un proyecto representativo de la unidad de Cuidado Familiar, priorizando la integración continua, automatización de pruebas, despliegue de soluciones e integración de CI/CD, evaluando el ajuste de la propuesta de estandarización del marco de trabajo de gestión de proyectos”, se concluye:

- La integración de pruebas automatizadas mediante *pytest* en el *pipeline* permite validar funcionalidades dentro del *workflow* con una cobertura del código del 16%, tomando en cuenta los procesos de la prueba de concepto, lo que se evidencia en la sección 5.3.3.
- Se evalúa con SonarQube el nivel de cobertura del *Branch* de *feature-standard-naming-1* correspondiente al 11,3%, identificando 2 errores críticos y 35 observaciones de mantenibilidad, evidenciado en la sección 5.3.3.
- El uso de Databricks Asset Bundles y GitHub Actions permite validar el *workflow*, *testing* y despliegue continuo, ejecutándose con un Gitflow en específico, tal como se evalúa en la sección 5.3.3.
- Es posible visualizar y monitorear el *pipeline* en GitHub Actions y Databricks, incorporando monitoreo por *logs*, duración de procedimientos y ejecución de métricas del *workflow*, tal como se evidencia en la sección 5.3.3.

- La prueba de concepto permite evidenciar la viabilidad técnica del marco DevOps CI/CD en procesos del equipo de Ingeniería de Inteligencia Artificial, mediante elementos de métricas como AI Readiness Criteria, tal como se muestra en la sección 5.3.3.

En primer lugar, se comprueba que la implementación del marco DevOps CI/CD permite una mejora significativa en la eficiencia operativa y del flujo del equipo de Ingeniería de Inteligencia Artificial. Esto se evidencia mediante la automatización del *pipeline*, la integración de herramientas como SonarQube y Pytest, así como a la reducción de los tiempos de despliegue en comparación con el proceso manual anterior del equipo.

Además, la estandarización de procesos contribuye a una mejor colaboración entre los miembros del equipo y a un control más riguroso de la calidad de las soluciones desplegadas. Se logró una alineación más clara entre los objetivos del equipo y los procedimientos técnicos utilizados. Finalmente, los resultados obtenidos validan la propuesta de estandarización como una solución viable y efectiva para optimizar la gestión de proyectos del equipo de Ingeniería de Inteligencia Artificial en la empresa manufacturera.

7. Recomendaciones

En este capítulo, se presentan las recomendaciones derivadas del análisis de resultados y de la validación de la propuesta de solución. Estas recomendaciones tienen como propósito guiar a la organización en la adopción efectiva del marco de trabajo propuesto en este trabajo, para la gestión eficiente del equipo de Ingeniería de Inteligencia Artificial.

Las sugerencias expuestas responden tanto a hallazgos técnicos como operativos, buscando fortalecer la colaboración, estandarización de procesos, calidad del desarrollo y fomentar la cultura de automatización y mejora continua.

7.1. Recomendaciones generales

A continuación, se describen las recomendaciones generales para la organización respecto a la gestión de proyectos de Ingeniería de Inteligencia Artificial, la estandarización de procesos y la adopción del marco DevOps CI/CD:

- Formalizar la integración del AI Readiness Criteria como prerrequisito para iniciar la planificación de cualquier proyecto, evaluando cuáles herramientas son necesarias para el desarrollo próximo.
- Destinar tiempo y recursos del equipo de Ingeniería de Inteligencia Artificial a la incorporación de prácticas de planificación, documentación, pruebas y despliegue que reflejen los lineamientos del AI Readiness Criteria, con la posibilidad de incorporar el ciclo de vida DevOps CI/CD.
- Implementar sesiones de retroalimentación entre los AI Engineers y los facilitadores de Databricks para evaluar el cumplimiento de estándares del AI Readiness Criteria que son implementables a mediano y largo plazo.
- Evaluar la asignación de un clúster exclusivo para el equipo de Ingeniería de Inteligencia Artificial que desarrolla proyectos para la Unidad de Cuidado Familiar, con el objetivo de optimizar costos y entregas alineadas a las métricas del AI Readiness Criteria y del marco de DevOps CI/CD.
- Establecer mecanismos de comunicación entre equipos técnicos y operativos, utilizando plataformas colaborativas como Confluence, Jira y GitHub para garantizar que todos los actores implicados accedan a la documentación técnica y operativa de forma actualizada.
- Promover una cultura de mejora dentro del equipo de Ingeniería de Inteligencia Artificial mediante DevOps CI/CD, con metodologías y métricas ágiles que desarrolle un cambio en el nivel de trabajos en producción del equipo.

7.2. Recomendaciones por objetivo

Objetivo	Recomendaciones
<p>Analizar las necesidades y limitaciones actuales del equipo de Inteligencia Artificial, en relación con el marco de trabajo actual, para identificar los requerimientos técnicos y operativos necesarios para implementar correctamente la metodología DevOps CI/CD dentro de los proyectos de la Unidad de Cuidado Familiar.</p>	<ul style="list-style-type: none"> • Realizar diagnósticos por cada año fiscal del porcentaje de proyectos en margen con el AI Readiness Criteria. • Documentar y actualizar formalmente las fases del ciclo de vida de los proyectos, tomando en cuenta requerimientos, procesos de desarrollo, modelo de datos, <i>testing</i> y despliegue. • Desarrollar una documentación estandarizada de los procedimientos necesarios para lograr la implementación de procesos como <i>code review</i>, GitFlow y prácticas DevOps que permitan realizar un <i>onboarding</i> efectivo en las métricas y lineamientos que se buscan estandarizar.
<p>Diseñar el marco de trabajo basado en la metodología DevOps CI/CD, según los lineamientos operativos y técnicos identificados, asegurando las mejores prácticas ágiles que sean adaptables al desarrollo de proyectos del equipo de Inteligencia Artificial.</p>	<ul style="list-style-type: none"> • Consolidar el uso obligatorio de herramientas del marco de trabajo como GitHub Actions, Confluence, pytest, SonarQube, DABs y su integración en todas las etapas del ciclo de vida de proyectos.
<p>Comparar las diferencias operacionales en términos de eficiencia, tiempos de entrega y calidad de las soluciones informáticas de la gestión de proyectos de Inteligencia Artificial entre la situación actual y la propuesta de estandarización basada en DevOps CI/CD, con el propósito de validar las mejoras implementadas en el ciclo de vida de los proyectos.</p>	<ul style="list-style-type: none"> • Establecer métricas que definan el uso adecuado de herramientas y procedimientos DevOps CI/CD que funcionen como un estándar para aplicar de forma efectiva esta metodología y su definición como punto de éxito. • Establecimiento de métricas de rendimiento como <i>Lead Time</i>, <i>Deployment Frequency</i>, <i>Change Fail Rate</i>, <i>Time to restore</i> para todos los proyectos del equipo.

Objetivo	Recomendaciones
<p>Elaborar una prueba de concepto en un proyecto representativo de la Unidad de Cuidado Familiar, priorizando la integración continua, automatización de pruebas, despliegue de soluciones e integración de CI/CD, evaluando el ajuste de la propuesta de estandarización del marco de trabajo de gestión de proyectos.</p>	<ul style="list-style-type: none">• Promover la implementación obligatoria de un <i>pipeline</i> automatizado con la documentación de todas las fases del ciclo de desarrollo, tal como se presenta en las figuras BPMN.• Realizar la prueba de concepto con proyectos que utilicen distintas tecnologías o metodologías a la presentada en la prueba de concepto.• Realizar monitoreos de desempeño cada vez que se lleva a cabo un despliegue del <i>pipeline</i>.

Fuente: elaboración propia.

8. Referencias

- Acosta Faneite, S. F. (2023). Los enfoques de investigación en las Ciencias Sociales. *Revista Latinoamericana Ogmios*, 3(8), 82–95. <https://doi.org/10.53595/rlo.v3.i8.084>
- Álvarez Carulla, A. (2021). Introducción al control de versiones (Git).
- Asana, T. (2025, February 18). Prueba de concepto (POC): qué es y cómo implementarla [2025] • Asana. Asana. <https://asana.com/es/resources/proof-of-concept>
- Atlassian. (n.d.). *Introducción a Jira: guía completa para principiantes* | Atlassian. <https://www.atlassian.com/es/software/jira/guides/getting-started/introduction#what-is-jira-software>
- Briceño, M. (2005). Guía para la elaboración de informes técnicos y tesis de grado. Mérida: Escuela de Ingeniería Química, Universidad de Los Andes, pp.1-2.
- Foqum Analytics, Empowers your success. (2023, December 11). *AI Engineering* | FOQUM. FOQUM. <https://foqum.io/blog/termino/ai-engineering/>
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* (1.^a ed.). IT Revolution.
- Freixas, J. (2022). La transformación digital de la mano del cloud computing y DevOps. Universitat de Barcelona, 25.
- Cabrera Méndez, M. (2010). Introducción a las fuentes de información.
- Cihan, U., Haratian, V., İçöz, A., Gül, M. K., Devran, Ö., Bayendur, E. F., ... & Tüzün, E. (2024). Automated Code Review In Practice. arXiv preprint arXiv:2412.18531.
- Digevo. (2024, August 6). *AI Factory: Una metodología para renovar a las industrias con Inteligencia Artificial*. <https://digevo.com/blog-categoria-optimizacion-de-procesos/ai-factory-metodologia-renovar-industrias-inteligencia-artificial/>

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Entender las GitHub Actions - Documentación de GitHub. (n.d.). GitHub Docs.

<https://docs.github.com/es/actions/about-github-actions/understanding-github-actions>

González, G. N. C., & Calderón, J. A. F. (2023). Integración y despliegue continuo con DevOps como cultura en empresas del sector TI colombianas. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, 10(19), 103-126.

Hernández-Sampieri, R., & Mendoza, C. (2020). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta.*

Introducción a la Investigación: guía interactiva. (s. f.).

<https://www.uv.mx/apps/bdh/investigacion/unidad1/investigacion-tipos.html>

Jobicy. (2025). *Data Engineer Salary in Costa Rica 2025 – Jobicy.*

<https://jobicy.com/salaries/cr/data-engineer#salary-section>

Kim, G., Humble, J., Debois, P., & Willis, J. (2021). *The DevOps Handbook: How to create world-class Agility, Reliability & Security in technology organizations* (2.^a ed.). IT Revolution Press.

Kinsman, T., Wessel, M., Gerosa, M. A., & Treude, C. (2021, May). How do software developers use github actions to automate their workflows? In 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR) (pp. 420-431). IEEE.

La integración y la distribución continuas (CI/CD). (s. f.).

<https://www.redhat.com/es/topics/devops/what-is-ci-cd>

Mamani Larico, B. N. (2020). Implementación de ambientes de integración y despliegue continuo (CI/CD) para aplicaciones empresariales en el centro de datos de la Universidad Peruana Unión Filial Juliaca.

Mendoza Torres, A. (2023). *Integración del proceso de diseño de User Experience al ciclo DevOps CI/CD* (Doctoral dissertation, Universidad de San Carlos de Guatemala).

Microsoft. (2023.). *¿Qué es Git? - Azure DevOps.* Microsoft Learn.

<https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>

Microsoft. (2025, 23 abril). What are Databricks Asset bundles? *Microsoft Learn.*

<https://learn.microsoft.com/en-us/azure/databricks/dev-tools/bundles/>

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa de manufactura mediante la metodología DevOps CI/CD

Obando, R. (2023, 21 enero). Qué es la estandarización de procesos, cómo aplicarla y ejemplos.

HubSpot. <https://blog.hubspot.es/sales/estandarizacion-de-procesos#:~:text=La%20estandarizaci%C3%B3n%20de%20procesos%20consiste,definidos%20de%20calidad%20y%20eficiencia>.

Okken, B. (2022). *Python testing with Pytest: simple, rapid, effective, and scalable*. The Pragmatic Programmers LLC.

Ospina Delgado, J. P. (2015). Análisis de seguridad y calidad de aplicaciones (Sonarqube).

Project Management Institute. (2021). *PMBOK Guide 7th edition*. <https://www.pmi.org/pmbok-guide>
[standards/foundational/pmbok?sc_campaign=D750AAC10C2F4378CE6D51F8D987F49D](https://www.pmi.org/pmbok-guide/standards/foundational/pmbok?sc_campaign=D750AAC10C2F4378CE6D51F8D987F49D)
¿Qué es DevOps? Explicación de DevOps | Microsoft Azure. (n.d.).

<https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>

¿Qué es la ingeniería de fiabilidad del sitio? - Explicación de la ingeniería de fiabilidad del sitio - AWS. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/sre/>

¿Qué es la seguridad de la información (InfoSec)? | Seguridad de Microsoft. (n.d.).

<https://www.microsoft.com/es-mx/security/business/security-101/what-is-information-security-infosec>

Tenempaguay, C. D. M. (2023). DevOps en el desarrollo de software: integración y entrega continua. *Polo del Conocimiento: Revista científico-profesional*, 8(9), 603-615.

Rodríguez, L. C., & Rodríguez, L. C. (2020, 4 julio). *sujeto de investigación*. Lizardo Carvajal R. https://www.lizardo-carvajal.com/sujeto-de-investigacion/#google_vignette

Sarango, A. H., Pallmay, E. R. C., Sarzosa, J. P. R., & Pozo, J. E. C. (2024). Tipos y clasificación de las investigaciones. *Latam: revista latinoamericana de Ciencias Sociales y Humanidades*, 5(2), 67.

Sdgilley. (2024, September 19). What is Azure Machine Learning? - Azure Machine Learning. Microsoft Learn. <https://learn.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning?view=azureml-api-2>

Van Merode, H. (2023). *Continuous Integration (CI) and Continuous Delivery (CD): A Practical Guide to Designing and Developing Pipelines* (1.^a ed.). <https://doi.org/10.1007/978-1-4842-9228-0>

What are Databricks Asset Bundles? | Databricks Documentation. (2025, January 15). <https://docs.databricks.com/gcp/en/dev-tools/bundles/>

Windgassen, C. (2025, March 25). *Guía de modelado, reglas y símbolos de BPMN 2.0* | BIC Software. Guía De Modelado, Reglas Y Símbolos De BPMN 2.0 | BIC Software. <https://www.gbtec.com/es/recursos/bpmn/>

9. Apéndices

En esta sección, se destacan los recursos adicionales del documento, que son de ayuda para la gestión de este.

9.1. Apéndice A. Plantilla de minutas

Logo de la empresa		MINUTA DE REUNIÓN	
Proyecto: Nombre exacto del mismo			
Reunión No.	Es un núm. consecutivo para este proyecto	Fecha:	Indicar la fecha exacta de la reunión
Lugar:	Indicar dónde fue la reunión	Hora Inicio/Finalización:	xx:00 am. / yy:00 am
Objetivo de la reunión:			
Participantes:	Presentes:		
	Ausentes:		
Temas Tratados			
No.	Asunto	Comentarios	Acuerdos
1	Debe ser detallado, explícito	Debe ser detallado, explícito	Debe ser detallado, explícito
2	Debe ser detallado, explícito	Debe ser detallado, explícito	Debe ser detallado, explícito
3	Debe ser detallado, explícito	Debe ser detallado, explícito	Debe ser detallado, explícito
Próxima reunión			
Temas a tratar		Fecha	Convocados
En la próxima reunión		indicar	Nombre de quiénes asistirán a esta próxima reunión.

Fuente: Escuela de Administración de Tecnología de Información.

9.2. Apéndice B. Plantilla de gestión de cambios

Hoja de Control de Cambios			
Datos Generales del Cambio			
Nº Cambio			
Solicitante		Fecha de solicitud del cambio	
Responsable de la implementación		Fecha de realización del cambio	
Estado	<input type="checkbox"/> Aprobado <input type="checkbox"/> En Revisión <input type="checkbox"/> Rechazado		
Detalles del Cambio			
Categoría	Introducción / Alcance / Marco Teórico / Metodología / ...		
Descripción detallada			
Justificación			
Implicaciones de realizar el cambio			
Impacto	Especificar si el cambio genera impacto en otras áreas del proyecto, tales como recursos, cronogramas, otros proyectos, entre otros.		
Comentarios/ Observaciones			
Revisado por:		Elaborado por:	
<u>Nombre tutor</u>		<u>Nombre estudiante</u>	
<u>Firma</u>		<u>Firma</u>	
(Prof. tutor)		(Estudiante)	
Revisado por:		Aprobado por:	
<u>Nombre representante empresa</u>		<u>Nombre Coordinadora TFG</u>	
<u>Firma</u>		<u>Firma</u>	
(Empresa)		(Coordinadora de TFG)	

Fuente: Escuela de Administración de Tecnología de Información.

9.4. Apéndice D. Plantilla de revisión documental

Revisión documental #	
Tipo de documento	
Autor	
Título	
Fuente en APA	
Palabras clave	
Descripción general del Documento	
Observaciones	

Fuente: elaboración propia.

9.5. Apéndice E. Plantilla de revisión documental interna en la organización

Revisión documental #	
Identificador de Revisión	
Fecha de revisión	
Nombre del documento	
Autor del documento	
Capítulo del documento analizado (Si aplica)	
Relación con la investigación.	

Fuente: elaboración propia.

9.6. Apéndice F. Plantilla para entrevista semiestructurada

Entrevista #			
Entrevistador		Fecha	
		Hora de inicio	
		Hora fin	
Entrevistado		Rol	
		Equipo	
Propósito			
Preguntas			
Observaciones			
Evidencia			

Fuente: elaboración propia.

9.7. Apéndice G. Entrevista N.º 1

Entrevista #1			
Entrevistador	Daniel Eduardo Obaldía Solís	Fecha	14/04/2025
		Hora de inicio	9:30 a.m.
		Hora fin	10:00 a.m.
Entrevistado	Mauricio Maroto	Rol	<i>AI Engineer</i>
		Equipo	<i>AI Engineering / Family Care</i>
Propósito	Recopilar información sobre la situación actual del equipo de Ingeniería de Inteligencia Artificial en la unidad de cuidado familiar.		
Preguntas			
Situación actual de la gestión de proyectos (V01)			
<u>Enfoque: procesos, herramientas, flujos, cuellos de botella, documentación y automatización</u>			
<ol style="list-style-type: none"> 1. Cómo describiría el proceso típico de desarrollo de un proyecto en su equipo o propio desde su inicio hasta el despliegue en producción. 2. ¿Qué herramientas se utilizan actualmente para planificar, documentar y hacer seguimiento de los proyectos? 3. ¿En qué etapas del ciclo de vida de proyectos considera que se presentan más cuellos de botella o retrasos? ¿Por qué? 4. ¿Se siguen procedimientos o metodologías estandarizadas para el desarrollo de los proyectos? 5. ¿Qué nivel de documentación se realiza actualmente en cada fase? (planificación, desarrollo, pruebas, despliegue). 6. ¿Qué tan automatizadas están las tareas clave como pruebas, despliegues o generación de documentación? 7. ¿Considera que existe una deuda técnica significativa en los proyectos actuales? ¿En qué áreas principalmente? ¿Considera que existe una acumulación de deuda técnica? 8. ¿Qué impacto y limitaciones tiene el <i>Launchpad</i> de <i>Databricks</i> para la gestión de los proyectos de ingeniería de inteligencia artificial? 9. ¿Podría describir brevemente cómo es el flujo de trabajo (<i>E2E</i>) de un proyecto de IA en el equipo? 10. ¿Existen procedimientos documentados o estandarizados para el desarrollo de proyectos? 			

Observaciones

1. Proceso típico E2E:

- a. El ciclo de trabajo inicia con la identificación del use case con el cliente y termina evaluando la viabilidad técnica, regulatoria y de costos.
- b. El procedimiento no es estandarizado formalmente, depende mucho del criterio del ingeniero y del cliente.

2. Herramientas utilizadas

- a. Las más comunes son Databricks, Outlook, Teams, Office y notes.
- b. Herramientas de gestión como *Jira* y *Confluence* no se usan sistemáticamente, salvo en un único proyecto.

3. Cuellos de botella

- a. Ubicados principalmente en la fase de definición del valor del proyecto con el cliente.
- b. La falta de claridad de parte del cliente puede ralentizar el inicio y diseño del proyecto.

4. Metodologías y procedimientos

- a. No existe una metodología estandarizada clara implementada (Hablando del ciclo devops). El flujo depende el del profesional y la confianza.

5. Documentación

- a. No hay documentación técnica formal ni sistematizada.
- b. Se generan presentaciones y resúmenes ejecutivos para clientes o negocio, no para el equipo técnico.

6. Automatización

- a. Se realizan pruebas de integración (Integration testing), pero no existen pipelines automatizados.
- b. Carencia de automatización de pruebas unitarias o despliegues continuos (CI/CD)

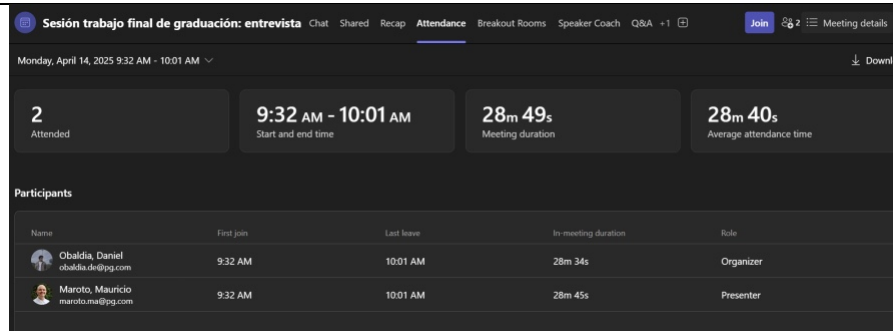
7. Deuda Técnica

- a. Si existe, especialmente en automatización, procesos manuales y estandarización de procesos
- b. El entrevistado expresó interés en recibir mayor soporte técnico para reducir el tiempo dedicado a tareas fuera de su perfil.

8. Flujo E2E

- a. No existe un flujo documentado ni herramientas de seguimiento continuo.

Evidencia



Name	First join	Last leave	In-meeting duration	Role
Obaldía, Daniel obaldia.de@ppg.com	9:32 AM	10:01 AM	28m 34s	Organizer
Maroto, Mauricio maroto.ma@ppg.com	9:32 AM	10:01 AM	28m 45s	Presenter

Fuente: elaboración propia.

9.8. Apéndice H. Entrevista N.º 2

Entrevista #2			
Entrevistador	Daniel Eduardo Obaldía Solís	Fecha	22/04/2025
		Hora de inicio	2:00 p.m.
		Hora fin	2:45 a.m.
Entrevistado	María Venegas	Rol	Senior AI Engineer
		Equipo	AI Engineering / Family Care
Propósito	Recopilar información sobre la situación actual del equipo de Ingeniería de Inteligencia Artificial en la unidad de cuidado familiar.		
Preguntas			
Situación actual de la gestión de proyectos (V01)			
<u>Enfoque: procesos, herramientas, flujos, cuellos de botella, documentación y automatización</u>			
<ol style="list-style-type: none"> 1. Cómo describiría el proceso típico de desarrollo de un proyecto en su equipo o propio desde su inicio hasta el despliegue en producción. (E2E) 2. ¿Qué herramientas se utilizan actualmente para planificar, documentar y hacer seguimiento de los proyectos? 3. ¿En qué etapas del ciclo de vida de proyectos considera que se presentan más cuellos de botella o retrasos? ¿Por qué? 4. ¿Cuáles considera que son los procedimientos o metodologías estandarizadas para el desarrollo de los proyectos? 5. ¿Qué nivel de documentación se realiza actualmente en cada fase? (planificación, desarrollo, pruebas, despliegue). 6. ¿Considera que existe una deuda técnica significativa en los proyectos actuales? ¿En qué áreas principalmente? ¿Considera que existe una acumulación de deuda técnica? 7. ¿Qué impacto y limitaciones tiene el <i>Launchpad</i> de <i>Databricks</i> para la gestión sus proyectos? 8. ¿Existen procedimientos documentados o estandarizados para el desarrollo de proyectos? 			
Cumplimiento de métricas y estándares de calidad (V02)			
<u>Enfoque: AI Readiness Criteria, métricas, calidad del código, cumplimiento normativo, limitaciones técnicas.</u>			
<ol style="list-style-type: none"> 1. ¿Aplica el conjunto de métricas definidas por el AI Readiness Criteria? Si en sus proyectos no se encuentra aplicado, ¿Cuáles serían las limitaciones que le impiden integrar estas prácticas en el flujo de trabajo concurrente? 2. ¿Cómo se asegura de la calidad del código durante el desarrollo? 3. ¿Qué tan integradas están actualmente las prácticas de confiabilidad, seguridad y sostenibilidad en los proyectos? 			

4. ¿Cuáles considera que son las principales limitaciones técnicas y operativas que enfrenta el equipo actualmente con estándares de calidad?
5. ¿Ha aplicado procedimientos de seguridad en sus proyectos, tomando en cuenta sostenibilidad o confiabilidad.?
6. ¿Qué automatizaciones con Inteligencia Artificial ha desarrollado en sus proyectos?

Integración con el equipo de SRE

Enfoque: Identificar limitaciones actuales y requerimientos para una colaboración más efectiva.

1. ¿Considera que existen limitaciones para que el equipo *SRE* adopte proyectos provenientes del equipo de *Family Care*?
2. ¿Qué tipo de requerimientos necesita actualmente *SRE* para brindar un soporte eficiente a los proyectos de AI en *Family Care*?

Observaciones

Evidencia

The screenshot shows a Zoom meeting attendance report for a session titled "Sesión trabajo final de graduación: entrevista" held on Tuesday, April 22, 2025, from 2:02 PM to 2:50 PM. The report includes the following statistics:

- 2 Attended
- Start and end time: 2:02 PM - 2:50 PM
- Meeting duration: 48m 21s
- Average attendance time: 48m 16s

The participants list is as follows:

Name	First join	Last leave	In-meeting duration	Role
Obaldía, Daniel obaldia.de@pg.com	2:02 PM	2:50 PM	48m 13s	Organizer
Venegas, Maria venegas.mi@pg.com	2:02 PM	2:50 PM	48m 19s	Presenter

Fuente: elaboración propia.

9.9. Apéndice I. Revisión documental interna de un proyecto en operación del equipo de Ingeniería de Inteligencia Artificial.

Revisión documental interna #1	
Identificador de Revisión	001
Fecha de revisión	28/04/2025
Nombre del documento	<i>Project Name: SRE Production Support Documentation.</i>
Autor del documento	Colaboradores del equipo de Ingeniería de Inteligencia Artificial.
Capítulo del documento analizado (Si aplica)	N/A
Relación con la investigación.	Esta documentación permite identificar prácticas formales que han sido efectivas dentro del contexto organizacional en el cual se desarrolla la propuesta. Este caso en particular representa un ejemplo de cómo debe de estructurarse la documentación técnica y operativa de un proyecto de la organización. A su vez, permite ver que estándares, herramientas y procedimientos son necesarios para la alineación con el negocio.

Fuente: elaboración propia.

9.10. Apéndice J. Revisión documental interna de un proyecto en operación del equipo de Ingeniería de Inteligencia Artificial desarrollado para la Unidad de Cuidado Familiar.

Revisión documental interna #2	
Identificador de Revisión	002
Fecha de revisión	29/04/2025
Nombre del documento	<i>Project Name: Proyecto análisis de datos de energía.</i>
Autor del documento	Colaboradores del equipo de Ingeniería de Inteligencia Artificial.
Capítulo del documento analizado (Si aplica)	N/A
Relación con la investigación.	La documentación sobre este proyecto desarrollado específicamente para la Unidad de Cuidado familiar, es de suma importancia para alinear e identificar prácticas técnicas y operativas para la creación del marco estandarizado <i>DevOps CI/CD</i> . Esto debido a que es un proyecto que se encuentra en desarrollo, pero a su vez esta transicionando a producción y a la búsqueda de un marco estandarizado para la gestión del código.

Fuente: elaboración propia.

9.11. Apéndice K. Revisión documental interna de las especificaciones del AI Readiness Criteria

Revisión documental interna #3	
Identificador de Revisión	003
Fecha de revisión	15/04/2025
Nombre del documento	<i>AI Readiness Criteria Standard Form and Requirements List.</i>
Autor del documento	Colaboradores del equipo de Ingeniería de Inteligencia Artificial.
Capítulo del documento analizado (Si aplica)	N/A
Relación con la investigación.	El <i>AI Readiness Criteria</i> representa un eje fundamental dentro de esta investigación ya que actúa como un conjunto de lineamiento para proyectos internos, el cual establece lo que debe cumplir una solución de Ingeniería de Inteligencia Artificial. A su vez, asegura el cumplimiento de políticas de TI, buenas prácticas de desarrollo, automatización, documentación y soporte operativo.

Fuente: elaboración propia.

9.12. Apéndice L. Revisión documental: *The DevOps Handbook 2021*.

Revisión Documental #1	
Identificador de Revisión	001
Fecha de revisión	15/04/2025
Nombre del documento	The DevOps Handbook: How to create world-class agility, reliability & security in technology organizations.
Autor del documento	Kim, G. , Humble, J. , Debois, P. & Willis, J. (2021).
Capítulo del documento analizado (Si aplica)	N/A
Relación con la investigación.	Esta obra es mundialmente conocida como un estándar para las prácticas DevOps ya que fundamenta teóricamente varios conceptos adoptados en el diseño del marco de trabajo propuesto, tal como la estructura de pipelines, estandarización de prácticas ágiles y la medición del rendimiento mediante métricas clave. Su inclusión valida la implementación de una cultura DevOps orientada a los equipos de desarrollo de software.

Fuente: elaboración propia.

9.13. Apéndice M. Revisión documental: *Continuous Integration and Continuous Delivery (2023)*

Revisión Documental #1	
Identificador de Revisión	002
Fecha de revisión	01/05/2025
Nombre del documento	Continuous Integration (CI) and Continuous Delivery (CD): A practical Guide to Designing and Developing Pipelines.
Autor del documento	Henry Van Merode
Capítulo del documento analizado (Si aplica)	N/A
Relación con la investigación.	Este libro ofrece una guía estructurada sobre la construcción de <i>pipelines</i> en entornos DevOps, abordando no solo aspectos técnicos sino también conceptuales como la integración continua, la entrega continua y su relación con arquitecturas modernas.

Fuente: elaboración propia.

9.14. Apéndice N. Revisión documental: *Accelerate: The Science of Lean Software and DevOps (2018)*

Revisión Documental #3	
Identificador de Revisión	003
Fecha de revisión	21/03/2025
Nombre del documento	
Autor del documento	
Capítulo del documento analizado (Si aplica)	
Relación con la investigación.	

Fuente: elaboración propia.

9.15. Apéndice O. Bitácora de reuniones.

07 de junio del 2025
Curridabat, San José

Escuela de Administración de Tecnologías de Información
Instituto Tecnológico de Costa Rica
TEC

Estimados señores (as):

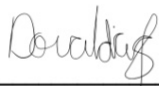
Por medio de la presente, hago constar que, durante el desarrollo del Trabajo Final de Graduación titulado "*Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa manufacturera mediante la metodología DevOps CI/CD*", se realizaron sesiones periódicas de seguimiento académico con el profesor tutor, Lic. Michael Sánchez Soto.

Estas sesiones tuvieron como propósito principal la revisión técnica, metodológica y de contenido del proyecto, permitiendo asegurar el cumplimiento de los entregables definidos, la alineación con los objetivos del trabajo y la incorporación de las observaciones brindadas por el tutor a lo largo del proceso.

Las minutas correspondientes a cada una de estas sesiones se encuentran adjuntas en este documento, que cubren desde el mes de marzo hasta junio del presente año.

Atentamente,

Daniel Eduardo Obaldía Solís
Estudiante
Licenciatura en Administración de Tecnologías de Información
Instituto Tecnológico de Costa Rica



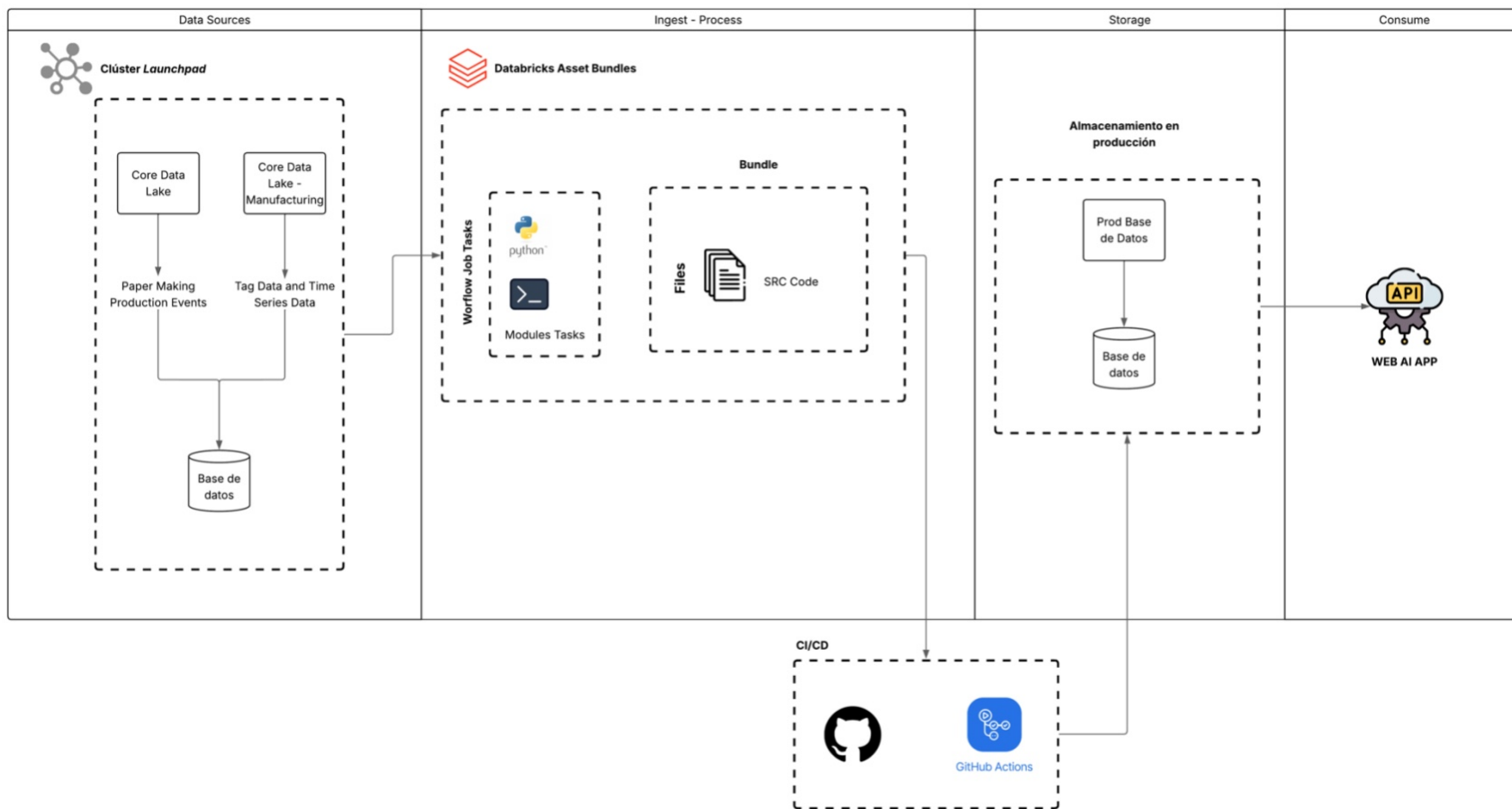
Daniel Eduardo Obaldía Solís
Estudiante
Licenciatura en Administración de Tecnologías de Información

MICHAEL
LIZANDRO
SANCHEZ SOTO
(FIRMA)

Digitally signed by
MICHAEL LIZANDRO
SANCHEZ SOTO (FIRMA)
Date: 2025.06.09
07:50:22 -06'00'

Lic. Michael Sánchez Soto
Profesor Tutor
Escuela de Administración de Tecnologías de Información

9.16. Apéndice P. Diagrama de Modelo de datos de un proyecto de la Unidad de Cuidado Familiar analizado en la documentación interna



Fuente: elaboración propia

10. Anexos

En esta sección, se visualizan los documentos que contienen información complementaria al documento, los cuales se identifican en la etapa de desarrollo del Trabajo Final de Graduación.

I. Glosario

Lista en orden alfabético de vocablos con sus respectivos significados, cuyo objetivo es lograr un adecuado entendimiento de este documento.

1. AI: corresponde al acrónimo en inglés de Inteligencia Artificial.
2. AI Engineering: se conoce como una disciplina que enfoca la aplicación de principios y prácticas de ingeniería para el desarrollo, implementación y mantenimiento de sistemas de Inteligencia Artificial.
3. AI Engineer: profesionales encargados de desarrollar, programar y entrenar algoritmos que funcionen con inteligencia artificial.
4. AI Factory: modelo organizativo y tecnológico que integra procesos y recursos para las soluciones basadas en IA.
5. AIE Readiness Criteria: cumplimiento de un conjunto de políticas, estándares y criterios diseñados para asegurar que los procesos y soluciones de TI sean robustas, seguras y operativamente correctas.
6. Azure Machine Learning (Azure ML): consiste en un servicio en la nube diseñado para acelerar y gestionar el ciclo de vida de proyecto de Machine Learning.
7. *Backlog*: lista priorizada de tareas, funcionalidades o requerimientos pendientes a desarrollar en un proyecto.
8. *Black*: librería de formateo automático de código para Python. Aplica reglas de estilo que aseguran la consistencia en la escritura del código.
9. *Branch*: rama del repositorio de GitHub que permite desarrollar nuevas funcionalidades, corregir errores o experimentar sin utilizar la versión estable del proyecto.
10. CI/CD: metodología de desarrollo de *software* que enfatiza en la integración continua de código en un repositorio de forma automática y periódica, lo que hace referencia a las siglas CI. Por su parte, las siglas CD es un proceso en el que se integran, prueban y se despliegan los cambios del código.

11. CLI (*Command Line Interface*): Herramienta de línea de comandos utilizada por desarrolladores para interactuar con sistemas operativos, repositorios y herramientas CI/CD. Facilita la ejecución y automatización de tareas, ejecución de scripts y flujos de trabajo.
12. *Code Review*: proceso de revisión del código por parte de otros desarrolladores o herramientas antes de su integración al repositorio principal.
13. Commit: Acción de registrar cambios en el código dentro de un repositorio de control de versiones, permitiendo documentar el avance de proyectos y facilitar la integración continua en el *pipeline CI/CD*.
14. *Continuous Improvement (CI)*: filosofía de mejora continua para la optimización de procesos y resultados mediante la implementación de pequeños cambios frecuentes, que buscan la excelencia operativa.
15. Core Data Lake (CDL): repositorio de datos centralizado que almacena grandes volúmenes de datos estructurados y no estructurados.
16. Copilot: asistente de Inteligencia Artificial desarrollado por Microsoft que facilita la revisión y generación de código dentro de *GitHub*,
17. Cuidado Familiar: término para referirse a la unidad de negocio de la empresa manufacturera.
18. Databricks: plataforma de análisis de datos que permite la integración de IA, *big data* y *machine learning*, facilitando la creación de soluciones computacionales.
19. Databricks Asset Bundles: herramienta diseñada para facilitar la adopción de buenas prácticas de ingeniería de *software* en proyectos de datos e inteligencia artificial. Mediante el uso de Databricks, esta herramienta permite facilitar la colaboración e implementación de procesos de CI/CD.
20. *Data Engineer*: profesional especializado en diseñar, construir y mantener infraestructura que permitan recopilar, almacenar, transformar y acceder a grandes volúmenes de datos.
21. Data Science: es una disciplina que combina conocimientos de estadística, programación y conocimiento de negocio para valorizar datos de la empresa.

22. DevOps: metodología que integra los equipos de desarrollo y operaciones para acelerar la entrega de *software* de alta calidad, fomentando la colaboración, la automatización y la mejora continua en todas las fases del ciclo de vida de los proyectos.
23. Estandarización: corresponde a la unificación de procedimientos, metodologías y operaciones dentro de una organización, con el fin de crear un marco de trabajo y cumplir con los criterios de calidad y eficiencia.
24. E2E (*End-to-End*): enfoque que abarca todas las fases de un proceso o sistema.
25. ETL: proceso de integración de datos que consiste en extraer, transformar y cargar la información.
26. *Features*: son funcionalidades que se implementan en un proyecto de *software*.
27. *Feedback*: es la información que se obtiene de una acción, proceso o comportamiento, y que es utilizada para ajustar o mejorar en ámbitos laborales.
28. Git: sistema de control de versiones que permite a los desarrolladores gestionar los cambios realizados en el código de un proyecto.
29. GitHub: plataforma de desarrollo colaborativo basada en Git, que permite gestionar versiones de código, colaboración en proyectos de *software* y automatizar proceso de CI/CD.
30. GitHub Actions: funcionalidad de la plataforma GitHub que permite automatizar flujos de trabajo directamente desde un repositorio, definiendo procesos como integración continua, entrega continua, despliegues, entre otros.
31. Hotfix/Rollback: corrección urgente aplicada directamente sobre un entorno productivo con el fin de resolver errores críticos.
32. IA: acrónimo de inteligencia artificial en español.
33. Infosec: conjunto de procedimientos y herramientas diseñadas para la protección de información confidencial de una empresa.
34. Infrastructure as a Code: consiste en la práctica que permite gestionar la infraestructura de TI mediante archivos de configuración, en lugar de procesos manuales.

35. Isort: librería de Python que gestiona la legibilidad del código y asegura la consistencia del estilo de este.
36. *Irisk*: Herramienta utilizada en la empresa para la evaluación de potenciales riesgos en las aplicaciones nuevas o modificadas. Esta herramienta se encuentra dentro de los lineamientos de *Infosec* y busca establecer pautas de seguridad, completitud y gestión de riesgos.
37. Lean: metodología de gestión enfocada en maximizar el valor para el cliente mediante la eliminación sistemática de desperdicios en los procesos.
38. LLM (*Large Language Model*): modelos de ciencias de datos para aprendizaje profundo que se reentrenan con grandes cantidades de datos.
39. Merge: Proceso de integración de cambios realizados en ramas de desarrollo o producción al Branch especificado, utilizado para consolidar nuevas funcionalidades o correcciones en proyectos de Ingeniería de Inteligencia Artificial.
40. Metodologías Ágiles: conjunto de prácticas y principios que promueven la colaboración y la entrega incremental de valor de proyectos de *software*.
41. Operacionalización: procedimiento de crear soluciones para entornos de producción mediante tecnologías y herramientas de esta índole.
42. Plan-Do-Check-Act: metodología de mejora continua utilizada para optimizar procesos. Consiste en cuatro fases: planificar, hacer, verificar y actuar.
43. Power BI: herramienta de análisis de datos de Microsoft.
44. Product Owner: Rol responsable de definir y priorizar los requerimientos del proyecto, asegurando que las soluciones de Inteligencia Artificial desarrolladas por el equipo generen valor para el negocio.
45. *Pipelines*: secuencia automatizada de pasos o proceso, que visualizan el código hasta un producto final, comúnmente utilizada en la metodología CI/CD.
46. Pull requests: Solicitudes de integración de cambios de código en el repositorio, las cuales permiten la revisión colaborativa (*code review*) antes de ser incorporadas al Branch principal, asegurando la calidad y estabilidad en el despliegue de proyectos CI/CD.

47. Scrum: marco de trabajo ágil utilizado en la gestión de proyectos.
48. Site Reliability Engineering (*SRE Engineering*): práctica que aplica principios de *software* que busca garantizar la confiabilidad y escalabilidad de sistemas.
49. SonarQube: plataforma de código abierto que permite realizar verificaciones o inspecciones de la calidad del código fuente mediante análisis estático.
50. SRE Engineer: utiliza herramientas de *software* para automatizar tareas de infraestructura de TI, como la administración de sistemas y el monitoreo de aplicaciones, asegurando la confiabilidad de estos.
51. Unit Testing: pruebas automatizadas que validan el correcto funcionamiento del código, permitiendo detectar errores de forma temprana y garantizando entregas de calidad dentro de *pipelines* de *CI/CD*.
52. Workflow: Flujo de trabajo estructurado que define la secuencia de tareas, herramientas y aprobaciones de un proyecto de Ingeniería de Inteligencia Artificial.
53. Yaml: Lenguaje utilizado para definir configuraciones en archivos legibles por humanos, comúnmente empleado para describir *pipelines* de *CI/CD* y *workflows* en herramientas como *GitHub Actions* y *Databricks Asset Bundles*.

II. Carta de Revisión Filológica

San José, 7 de junio de 2025

Señores(as)
Escuela de Administración de Tecnologías de Información
Instituto Tecnológico de Costa Rica

Estimados señores(as):

Yo, María Fernanda Sanabria Coto, cédula de identidad 114290780, filóloga española graduada en la Universidad de Costa Rica, perteneciente a la Asociación Costarricense de Filólogos (ACFIL), carné 225 y al Colegio de Licenciados y Profesores en Letras, Filosofía, Ciencias y Artes de Costa Rica (COLYPRO), código 75402, hago constar que he revisado el documento titulado:

Propuesta de estandarización del marco de trabajo para la gestión de proyectos de ingeniería de inteligencia artificial en una empresa manufacturera mediante la metodología DevOps CI/CD

Dicho documento fue elaborado por Daniel Eduardo Obaldía Solís, cédula de identidad 118060522, con el fin de optar al grado de Licenciatura en Administración de Tecnologías de Información. He revisado y corregido aspectos tales como construcción de párrafos, vicios del lenguaje trasladados a lo escrito, ortografía, puntuación y otros relacionados con el campo filológico.

Atentamente,

Fernanda S. Coto

María Fernanda Sanabria Coto
Filóloga
Cédula de identidad 1-1429-0780
Colypro. Código 75402
Asociación Costarricense de Filólogos. Carné nro. 225
fernanda.sanabria@filologos.cr
Teléfono: +506 6022 9569



MARIA
FERNANDA
SANABRIA
COTO
(FIRMA)

Firmado digitalmente por MARIA
FERNANDA SANABRIA COTO
(FIRMA)
Nombre de reconocimiento (DN):
serialNumber=CPF-01-1429-0780,
sn=SANABRIA COTO,
givenName=MARIA FERNANDA,
c=CR, o=PERSONA FISICA,
ou=CIUDADANO, cn=MARIA
FERNANDA SANABRIA COTO
(FIRMA)
Motivo: Revisión filológica
Ubicación: Costa Rica
Fecha: 2025.06.07 00:11:50 -06'00'