# Framework de MLOps para Equipos de Trabajo de pocas personas

Eduardo Jopia Fernández
eduardo.jopia@sansano.usm.cl

Oscar Mendoza Chávez
omendoza@usm.cl

Nicolás Sepúlveda Alvear
nsepulvedaa@usm.cl

**Abstract**

Las diferentes aplicaciones de *Machine Learning* (ML) en las actividades de las organizaciones han tenido un incremento explosivo durante los últimos años, así como la cantidad de profesionales, de diferente *background*, que se capacitan para poder desarrollar sus propios modelos. Sin embargo, se ha podido evidenciar que existen retos para implementar estos modelos en producción, y la norma más que la excepción es que los equipos de trabajo que gobiernan estas iniciativas son pequeños. Este artículo tiene como objetivo la revisión bibliográfica de diferentes propuestas de MLOps considerando las limitaciones de RRHH en las organizaciones.

**Keywords**

Machine Learning, MLOps, DevOps, Workflow Orchestration, Data Teams

## 1 Presentación del tema

### 1.1 Motivación

La elección del tema se centra en la creciente necesidad de efectivizar el alcance de los modelos de Machine Learning en la industria. A pesar de los avances en el desarrollo de estos modelos, muchas organizaciones enfrentan desafíos al llevarlos a un entorno de producción real. La transición de un modelo desde su fase de desarrollo hasta su implementación en producción es un proceso complejo que va más allá de la simple codificación. Implica integrar sistemas, coordinar equipos con diversas habilidades y garantizar que el modelo funcione de manera óptima en un entorno productivo. Esta complejidad subyacente y la necesidad de un enfoque estructurado para abordarla es uno de los pilares detrás de este documento.

Por otro lado, es importante destacar que la integración de este marco de trabajo no se limita solo a herramientas tecnológicas; sino que también debe considerar a las personas involucradas en el proceso, sus conocimientos y antecedentes, sus responsabilidades y sus motivaciones. Su comprensión y aceptación del proceso son esenciales para garantizar que los esfuerzos de los equipos desarrolladores e integradores de los Sistemas de Machines Learning sean fructíferos.

Finalmente, como equipo hemos podido identificar dos factores comunes que dificultan la puesta en producción de modelos de ML, y por lo tanto, que imposibilitan la obtención de valor en los productos de datos basados en ML: a) los tamaños de equipos que gobiernan las iniciativas de ML y

b) el background de conocimiento de las personas que desarrollan los modelos de ML y que dificulta el entendimiento del framework de ingeniería de Software de DevOps.

## 1.2 Objetivos y alcance

El objetivo principal es identificar los roles críticos y diseñar un Roadmap de Implementación de MLOps, que sirva como guía para las organizaciones, especialmente para equipos con recursos y tiempo limitados. Este marco referencial tiene como meta facilitar la coordinación entre los diversos actores involucrados, desde científicos de datos hasta ingenieros de software y equipos de DevOps. Además, se busca definir un pipeline claro y efectivo que cubra todas las fases, desde el desarrollo y prueba hasta el despliegue y monitoreo de soluciones basadas en ML.

## 1.3 Importancia del tema

MLOps ha emergido como una disciplina esencial en el ámbito del Machine Learning y la inteligencia artificial. Su relevancia radica en su capacidad para cerrar la brecha entre el desarrollo de modelos y su implementación en producción. En un mundo donde la rapidez y la adaptabilidad son cruciales, las organizaciones no pueden permitirse desarrollar modelos que no se traduzcan en soluciones prácticas y efectivas. Además, con el entorno tecnológico en constante evolución, es vital que las organizaciones tengan un enfoque flexible y escalable para MLOps, permitiéndoles adaptarse a nuevas herramientas y tecnologías a medida que surgen.

Es fundamental subrayar que la formación y el desarrollo profesional continuo, junto con una cultura organizacional que promueva la colaboración y el aprendizaje, son esenciales para garantizar que las organizaciones no solo adopten MLOps, sino que también lo hagan de manera efectiva y sostenible.

En el contexto de la minería en Chile, la integración de modelos de ML apunta directamente hacia la innovación y la optimización de los recursos. Proporcionar una solución tecnológica y operativa que permita tomar decisiones asertivas en tiempo real no solo promoverá el crecimiento y desarrollo del país, sino que también garantizará la seguridad de las personas y la sustentabilidad del medio ambiente. Por otra parte, la adopción de modelos de ML por parte de las instituciones financieras tiene como objetivo fortalecer la toma de decisiones al aprovechar el valor añadido generado por una variedad de modelos de aprendizaje automático, al tiempo que mejora la experiencia de sus productos y servicios.

# 2 Marco conceptual

Para llevar a cabo esta investigación, resulta fundamental establecer definiciones precisas de conceptos clave que ayudarán a comprender la importancia de este marco de trabajo. A continuación, se presentan los conceptos que se abordarán:

## 2.1 El Ciclo de Vida de un Proyecto de Machine Learning

El desarrollo de proyectos en el ámbito del aprendizaje automático engloba etapas esenciales de su ciclo de vida **[1]**, cada una crítica para el éxito del proyecto. Esta investigación bibliográfica ha permitido identificar y organizar dichas fases en categorías claves que facilitan la comprensión de su complejidad y dinamismo.

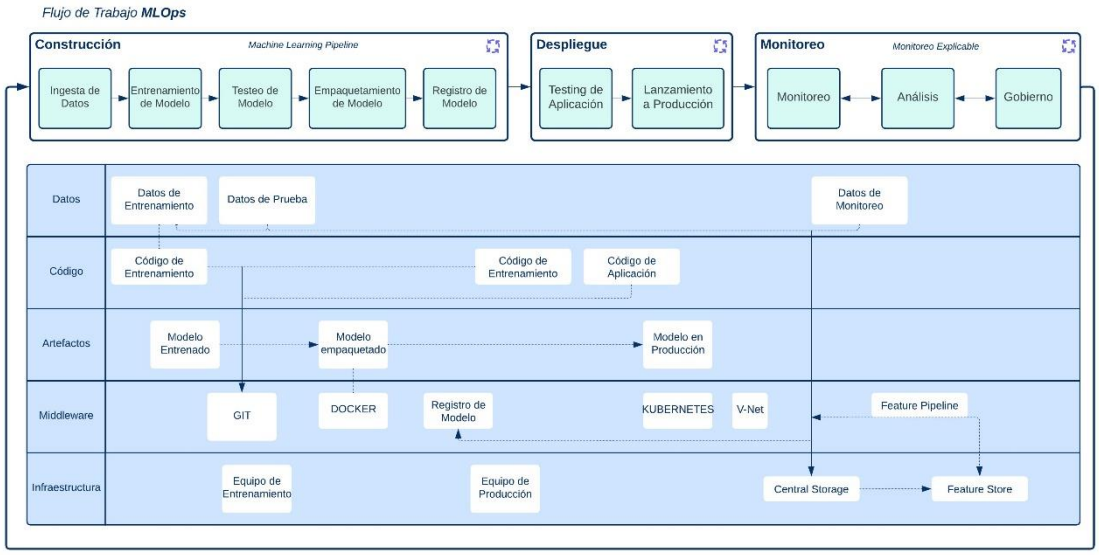

| Etapa | Características |
|---|---|
| Comprensión y Planificación del Proyecto | La fase inicial define los requisitos del proyecto con la colaboración entre expertos y científicos de datos, estableciendo objetivos y evaluando los datos para dirigir el proyecto. |
| Adquisición de Datos | En la etapa inicial de un proyecto de aprendizaje automático, se recopilan datos de diversas fuentes, evaluando su calidad y volumen para garantizar su adecuación para futuras predicciones. |
| Preparación y Exploración de Datos (EDA) | En esta fase, se limpian y transforman los datos, tratando problemas como valores faltantes y duplicados, y se normalizan para el modelado. Este proceso iterativo refina los datos para su idoneidad en análisis posteriores. |
| Modelado | La fase central implica seleccionar y entrenar el algoritmo óptimo, ajustando hiperparámetros y características para maximizar la precisión del modelo |
| Despliegue | Integrar y desplegar el modelo en producción para ejecutar predicciones en situaciones reales marca la conclusión práctica del proyecto. |
| Monitoreo y Mantenimiento | En el Post-despliegue, es vital supervisar y ajustar el rendimiento del modelo, reentrenándolo conforme cambien los datos. |

Esta estructuración no solo destaca la importancia de cada fase dentro del ciclo de vida de un proyecto de aprendizaje automático, sino que también subraya la necesidad de un enfoque iterativo y flexible para superar los desafíos inherentes a estas tecnologías. La colaboración constante entre distintos expertos y una evaluación continua de los procesos y resultados son fundamentales para el éxito de proyectos en este campo dinámico y en rápida evolución.

## 2.2 El flujo de MLOps

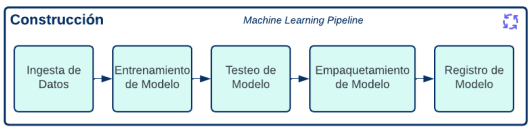
El flujo de trabajo de MLOps **[2] [3] [4]** representa una fusión sinérgica de prácticas provenientes de la ingeniería de datos, el aprendizaje automático (ML) y DevOps, constituyendo un paradigma modular y flexible para el despliegue y mantenimiento eficiente de soluciones de ML en entornos productivos. El proceso de MLOps se desglosa en dos componentes principales: el **pipeline** de MLOps y los **drivers** de MLOps, revelando cómo cada elemento contribuye al ciclo de vida de las soluciones de ML.

El Diagrama muestra el Flujo de MLOps



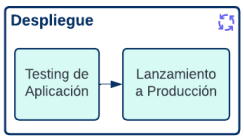

El pipeline como tal se constituye de tres fases: Construcción, Despliegue y Monitoreo.

### 2.2.1 Pipeline: Construcción



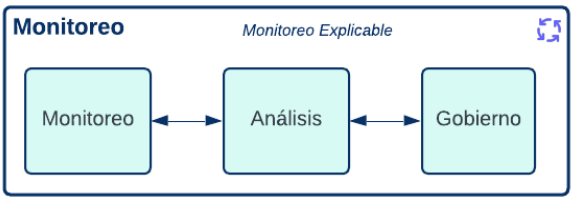

| **Definición** |
| --- |
| En esta fase, se diseñan y modelan los modelos de ML, seguido de la ingesta y preparación de datos para el entrenamiento. Posteriormente, se ajustan los hiperparámetros para afinar el rendimiento del modelo, que luego se evalúa, empaqueta y versiona preparándolo para el despliegue. |

### 2.2.2 Pipeline: Despliegue



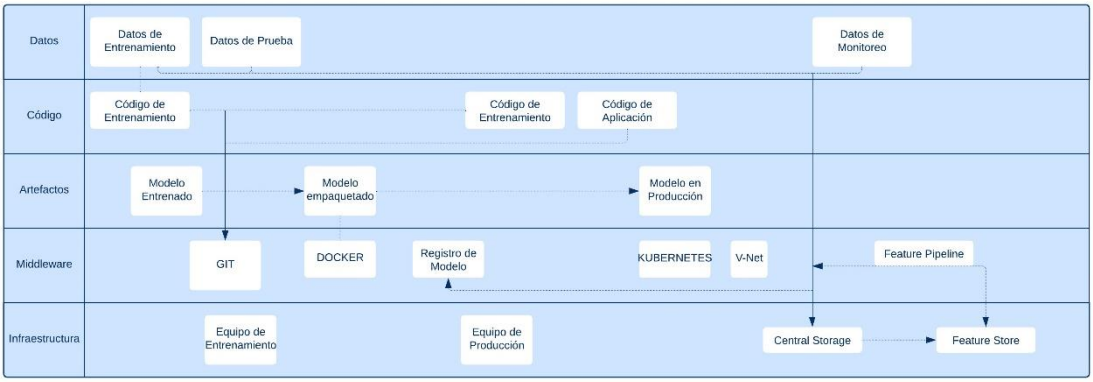

| **Definición** |
| --- |
| Una vez listo, el modelo se despliega en producción para probar su efectividad y escalabilidad. |

### 2.2.3 Pipeline: Monitoreo




| Definición |
|---|
| Esta etapa se enfoca en monitorear continuamente el modelo en uso, evaluando su rendimiento y estabilidad mediante métricas y análisis de explicabilidad para informar decisiones de mantenimiento o mejora. |

### 2.2.4 Drivers




Los drivers o impulsores incluyen datos, código, artefactos, middleware e infraestructura, fundamentales para habilitar y optimizar el pipeline de MLOps:

| Drivers | Descripción |
|---|---|
| **Datos** | Los datos son esenciales para entrenar y evaluar modelos de ML, abarcando desde la adquisición y preparación hasta su ampliación para diversas fases del pipeline. |
| **Código** | Incluye código para entrenar y probar modelos de ML, desarrollando aplicaciones relacionadas, gestionado con control de versiones y automatizado mediante CI/CD. |
| **Artefactos** | Incluye scripts y código para entrenar, probar modelos de ML y desarrollar aplicaciones, manejado con control de versiones y automatizado con CI/CD. |
| **Middleware** | Este apartado trata sobre la creación y gestión de código para entrenar, testear modelos de ML y desarrollar aplicaciones, utilizando control de versiones y CI/CD para automatización. |
| **Infraestructura** | Los recursos de computación y almacenamiento son vitales para entrenar, probar e implementar modelos de ML, disponibles tanto en entornos locales como en la nube, según las necesidades del proyecto. |

La implementación de un flujo de trabajo de MLOps automatizado y bien optimizado ofrece ventajas significativas, como la mejora en la eficiencia de los equipos de TI y la reducción de costos operacionales, siendo esencial para las organizaciones que buscan incorporar soluciones de ML en sus operaciones cotidianas.

## 2.3 Líderes Tecnológicos en Chile

Dado a lo nuevo de este tema, existen pocas instituciones las cuales se especializan en este marco de trabajo de las pocas que pudimos encontrar tenemos las siguientes:

- **BackSpace(**https://backspace.cl/**):** Empresa fundada en 2011, especializada en automatización de procesos y análisis avanzado. Atienden a varios sectores como seguros, banca, energía, y más.
- **Neuralworks**(https://neuralworks.cl/): Se especializan en desafíos técnicos como modelos predictivos y tecnología de detección de rostros, trabajando en proyectos internos y con corporaciones para crear productos digitales.
- **Soluciones Data & Analytics (**https://soluciones.cl/**):** Consultora con más de 20 años de experiencia en proyectos de Data & Analytics. Su equipo de consultores multidisciplinarios entrega soluciones tecnológicas personalizadas, utilizando una metodología flexible y adaptable.

## 2.4 Roles en el ciclo de vida de MLOps

En el dinámico campo del Machine Learning (ML) y su integración en procesos operativos a través de MLOps, la colaboración entre diversos roles profesionales **[5]** se convierte en un pilar fundamental para el éxito. Nuestra investigación abarca una revisión bibliográfica, complementada con análisis de casos prácticos, para identificar roles críticos que emergen como esenciales en el desarrollo e implementación efectiva de soluciones de ML en el entorno empresarial. A continuación, presentamos una descripción de los roles más comunes, resaltando sus responsabilidades principales.

| Rol | Responsabilidades |
|---|---|
| Data Scientist | Desarrolla modelos que atiendan las necesidades comerciales, facilitando su uso operacional en producción. La calidad del modelo se evalúa con expertos para asegurar que cumple con los requisitos comerciales establecidos. |
| Data Engineer | Optimice la recuperación y el uso de datos para impulsar los modelos de aprendizaje automático. |
| Ingenieros de Software | Integrar modelos de ML en los sistemas empresariales, asegurando su compatibilidad y funcionamiento fluido con aplicaciones no basadas en ML. |
| Machine Learning Engineer | Construir y evaluar sistemas operativos para seguridad, rendimiento y disponibilidad, y gestionar flujos de trabajo de CI/CD. |
| Data Architect | Asegurar un entorno de ML escalable y flexible para diseño, desarrollo y monitoreo, e incorporar nuevas tecnologías para optimizar el rendimiento de modelos de ML en producción. |

## 2.5 Tecnología

El despliegue de MLOps en una organización requiere del uso extensivo de tecnología **[6]**. El hecho de contar con un equipo pequeño de profesionales acentúa aún más esta dependencia. La siguiente es una lista que muestra opciones de herramientas que son utilizadas en las diferentes etapas del ciclo de MLOps:

| Etapa | Tecnología Requerida |
|---|---|
| Construcción | Lenguajes de Programación: Python, R<br>Librerías: Pandas, Numpy, TensorFlow, PyTorch, Keras<br>IDEs: VSC, Anaconda, Jupyter Notebook, PyCharm<br>Versionamiento de Código: Git.<br>Procesamiento de Datos: Apache Spark, Databricks, dbt.<br>Orquestación: Apache Airflow, GCP Dataflow, Azure Data Factory, AWS Glue.<br>Almacenamiento: Bases de Datos SQL y No-SQL, Cloud Storage (Google Cloud Storage, AWS S3, Azure Data Lake Gen 2.)<br>Entrenamiento: MLflow tracking<br>Infraestructura de Ambientes: Desarrollo, Testing/QA |
| Despliegue | Empaquetamiento: MLflow Models<br>Despliegue: Terraform, Jenkins<br>Containerización: Docker, Kubernetes |
| Monitoreo | Registro de Modelos: MLflow Model Registry<br>Monitoreo: Prometheus, Grafana |

## 2.6 Modelos de Madurez

El modelo de madurez de MLOps **[7] [8]** es una herramienta estratégica diseñada para ayudar a las organizaciones a evaluar y avanzar en sus capacidades de operaciones de Machine Learning. Proporciona un camino claro para la evolución desde prácticas ad hoc hasta procesos altamente automatizados y eficientes. Este modelo, creado por Microsoft, estructura en cinco niveles distintos de madurez, cada uno reflejando una etapa de desarrollo en la capacidad de implementar, gestionar y escalar soluciones de ML en producción.

*Niveles del Modelo de Madurez de MLOps*

| Nivel | Características | Tecnología |
|---|---|---|
| 0 – No MLOps | Este nivel se caracteriza por la manualidad en los procesos, la falta de colaboración entre equipos y un manejo local de modelos de ML, resultando en ineficiencias y problemas para escalar o gestionar el ciclo de vida de los modelos. | La construcción, testing, control de perfomance y entrenamiento del modelo y la aplicación se lleva a cabo de forma manual. |
| 1 - DevOps pero no MLOps: | Introduce automatizaciones en el proceso de lanzamiento de software, pero aún carece de integración y automatización específica para ML. Este nivel señala el inicio de la adopción de prácticas de DevOps, aunque los ciclos de vida de los modelos de ML dependen mucho de la intervención manual. | Los builds ya son automatizados, los tests para el código de aplicación son automatizados. |

| | | |
|---|---|---|
| 2 - Entrenamiento Automatizado | Este nivel incorpora automatización en el entrenamiento y gestión de modelos, señalando avances hacia prácticas de MLOps más maduras. Aunque el despliegue es manual, esta automatización mejora la experimentación y reproducción de modelos. | Entrenamiento automatizado, tracking centralizado de la performance del entrenamiento del modelo, gestión y administración de modelos. |
| 3 - Implementación de Modelo Automatizada | Este nivel logra una automatización completa en el lanzamiento, incluyendo pruebas e integración en producción, con enfoque en la trazabilidad y gestión de versiones, mejorando la agilidad y control del ciclo de vida de modelos de ML. | A/B testing de la performance del modelo para despliegue, tests automatizado para todo el código. |
| 4 - Operaciones Automatizadas de MLOps Completas | Este nivel es el pico de madurez, con un sistema totalmente automatizado para gestionar y desplegar modelos, y facilita la mejora continua. Ofrece un ciclo de vida de ML ágil y adaptativo, respondiendo a cambios en las necesidades del negocio y del entorno de datos. | Testeo y Entrenamiento automático.<br>Métricas centralizadas para el despliegue. |

### 2.7 Aspectos Éticos

Los aspectos éticos de MLOps están directamente relacionados con los aspectos éticos de la gestión de datos **[9]**, la seguridad y privacidad de los datos **[10]** y, particularmente, los posibles sesgos que pueden ocurrir durante el entrenamiento. Dentro de la gestión de datos, resaltan la responsabilidad organizacional en el proceso de desarrollo y la transparencia en el uso de datos.

En términos de Seguridad y Privacidad, los procesos de desarrollo deben ser gestionados incorporando los principios de Seguridad de la Información y, dependiendo la industria, los procesos de privacidad y seguridad demandados (como PCI-DSS) *considerando* que es probable que los profesionales detrás del desarrollo de modelos no tengan los conocimientos de seguridad suficientes como un desarrollador de software.

Finalmente, se deben establecer prácticas para asegurar que los algoritmos implementen sesgos por los datos de entrenamiento usados, ni tengan impactos negativos no intencionados. Artículos académicos y de investigación, como los publicados en "Ethics and Information Technology", abordan estas cuestiones, ofreciendo marcos y directrices para la conducta ética en IA y ML.

## 3 Análisis de Resultados

Nuestra investigación basada en una extensa revisión bibliográfica, destaca importantes desafíos en el campo del Machine Learning (ML) y cómo las prácticas de Machine Learning Operations (MLOps) ofrecen soluciones efectivas para enfrentarlos. En los resultados clave que presentaremos, se exponen estrategias y soluciones concretas que MLOps proporciona para superar

obstáculos típicos, respaldadas por prácticas recomendadas identificadas durante nuestro análisis.

Este resumen pretende equipar a profesionales y aficionados del ML con un conjunto de herramientas basadas en MLOps, trazando una ruta hacia el éxito en proyectos de ML. Abordaremos los principales problemas encontrados en todas las fases del ciclo de vida de ML, demostrando cómo MLOps actúa como un catalizador para la eficiencia y cohesión operativa. Además, se discutirán los roles esenciales en MLOps y se presentará un plan de implementación enriquecido con la adición de tecnologías recomendadas para la adopción de MLOps, ofreciendo a las organizaciones una guía práctica para adoptar y optimizar estas prácticas efectivamente.

### 3.1 Problemas identificados en el ciclo de vida de Modelos de Machine Learning y solución a través de MLOps

Nuestro estudio bibliográfico sobre el ciclo de vida de los proyectos de Machine Learning (ML) ha revelado una serie de desafíos fundamentales **[]** que oscilan desde la concepción inicial del proyecto hasta su despliegue y mantenimiento continuo. Estos desafíos no solo obstaculizan la eficiencia y eficacia de los proyectos de ML, sino que también comprometen su escalabilidad y sostenibilidad. Sin embargo, la implementación de prácticas de Machine Learning Operations (MLOps) emerge como una solución poderosa, ofreciendo estrategias para superar estos retos de manera efectiva. La siguiente tabla ilustra cómo MLOps aborda estos desafíos clave en el desarrollo de ML, ofreciendo soluciones para mejorar estas problemáticas

| **Etapa del Ciclo de vida de ML** | **Problemática Identificada** | **Solución vía MLOps** |
|---|---|---|
| **Comprensión y Planificación del Proyecto** | Brecha de Conocimiento y Comunicación entre Equipos: *Comunicación limitada entre equipos científicos y operativos.* | La integración y automatización en MLOps mejoran la comunicación y comprensión mutua entre los equipos. |
| **Adquisición de Datos** | Involucramiento en Actividades No Esenciales: *Científicos de datos dedican hasta un 40% de su tiempo en tareas periféricas como la limpieza de datos y gestión.* | Automatización en MLOps para agilizar la gestión de datos, permitiendo a los científicos de datos centrarse en el desarrollo y mejora de modelos. |
| **Preparación y Exploración de Datos (EDA)** | Manejo Ineficiente de la Versión y Reproducibilidad: *Falta de sistemas adecuados para el versionado complica la gestión de cambios.* | Sistemas de control de versiones y herramientas de seguimiento centralizado en MLOps garantizan reproducibilidad y trazabilidad. |
| **Modelado** | Código no Robusto ni Escalable: *El código desarrollado inicialmente a menudo no es apto para entornos de producción.* | Implementación de infraestructuras y middleware adecuados dentro del flujo de trabajo de MLOps para asegurar código robusto y escalable. |

| | | |
|---|---|---|
| **Despliegue** | Transición desde el Desarrollo al Despliegue: *Aproximadamente el 80% de los modelos de ML no superan la fase de prueba de concepto.* | Implementación de un pipeline de MLOps estandarizado para facilitar la transición eficiente de los modelos desde el desarrollo hasta la producción. |
| **Monitoreo y Mantenimiento** | Desafíos de Monitoreo y Mantenimiento Post-Despliegue: *La supervisión manual post-despliegue puede resultar en respuestas lentas a problemas.* | La supervisión automatizada en MLOps facilita la detección y solución rápida de problemas. |
| **Transversal** | Falta de Estandarización en el Desarrollo de Modelos: *La falta de procesos estandarizados limita la eficiencia y escalabilidad.* | Estandarización de procesos a través del pipeline de MLOps promoviendo prácticas uniformes en las etapas de desarrollo y despliegue. |
| **Transversal** | Automatización Insuficiente: *La limitada automatización en el ciclo de vida del ML ocasiona procesos lentos.* | Los flujos de trabajo de MLOps altamente automatizados agilizan desde el entrenamiento hasta el reentrenamiento de modelos. |
| **Transversal** | Reescritura y Repetición de Código: *La necesidad de adaptar código para producción introduce errores y retrasos.* | Prácticas de CI/CD en MLOps disminuyen la necesidad de reescribir código, optimizando el desarrollo y despliegue. |
| **Transversal** | Deuda Técnica en Proyectos de ML: *Acumulación de deuda técnica por falta de estandarización y automatización.* | La estandarización y automatización mediante MLOps minimizan la deuda técnica, mejorando la escalabilidad y mantenimiento. |

La integración de MLOps en el ciclo de vida de los proyectos de ML no es solo una mejora operativa; es una transformación estratégica que promueve la eficiencia, la colaboración y la innovación. Al adoptar un marco de trabajo de MLOps, las organizaciones pueden asegurar una implementación más efectiva y una operación más eficiente de soluciones de ML, resultando en proyectos más exitosos y sostenibles

## 3.2 Roles Recomendados

La selección de roles propuestos busca garantizar una operatividad fluida y cohesiva a lo largo de todas las fases del proceso de MLOps, promoviendo una colaboración efectiva y el desarrollo de soluciones robustas y escalables de ML. Definimos, por lo tanto, el equipo mínimo requerido para poder implementar MLOps y las responsabilidades de este se detallan a continuación:

| **Rol** | **Responsabilidades** |
|---|---|
| Ingeniero de DevOps (DevOps Engineer) | Administra la infraestructura esencial para el despliegue y operación de aplicaciones y modelos de ML, y gestiona las canalizaciones de CI/CD para automatizar pruebas, construcción y despliegue. |

| | |
|---|---|
| Ingeniero de Datos (Data Engineer) | Optimiza la gestión de datos para modelos de ML, incluyendo su recopilación, almacenamiento y procesamiento eficientes. Diseña y mantiene la infraestructura de datos para un análisis eficaz de grandes volúmenes de datos. |
| Ingeniero de ML (ML Engineer) | Diseña arquitecturas de soluciones de ML integradas y escalables. Gestiona el ciclo completo de vida de los modelos de ML, desde el versionado hasta el monitoreo y despliegue. |
| Científico de Datos (Data Scientist) | Desarrolla modelos de ML para abordar necesidades comerciales específicas. Evalúa la calidad y efectividad de los modelos junto a expertos del dominio. |

Esta estructura de equipo permite abordar de manera eficiente y efectiva los retos que surgen a lo largo de todo el ciclo de vida de desarrollo de soluciones de ML, desde la concepción hasta la implementación en entornos de producción, facilitando el desarrollo de proyectos de ML exitosos y de alto impacto comercial.

## 3.3 Tecnología

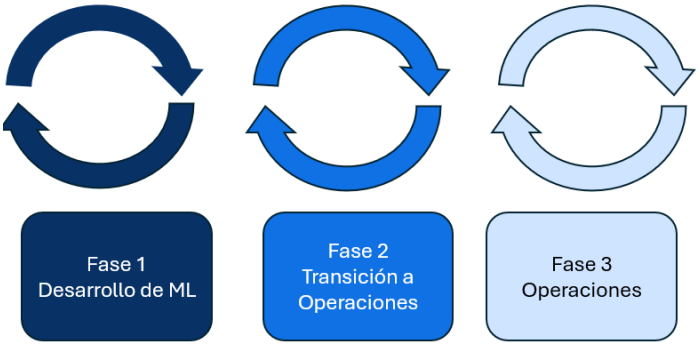
Para ejemplificar la implementación de una solución MLOps homogénea y eficaz **[11] [12]** se plantea la utilización de Google Cloud Platform (GCP) como el servicio en la nube y GitLab CI/CD para automatizar el flujo de trabajo en los ambientes DEV, QA y PROD A continuación, se detalla una propuesta para configurar estos ambientes:

| **Ambiente** | **Objetivos** | |
|---|---|---|
| **DEV** | Facilitar la experimentación rápida y el desarrollo de modelos de ML | |
| **QA** | Validar la calidad, rendimiento y estabilidad de los modelos antes de su despliegue en producción. | |
| **PRD** | Despliega modelos de ML estables y de alto rendimiento para inferencia o procesamiento por lotes, garantizando su disponibilidad, escalabilidad y seguridad en el entorno de producción. | |
| **Herramientas** | **Vertex AI** | Espacio de Trabajo para el Desarrollo de Modelos de ML |
| | **Jupyter Notebook** | Utiliza librerías de Python como Anaconda, pandas, numpy y sklearn para experimentar con modelos de forma eficiente y efectiva. |
| | **Airflow** | Orquesta pipelines de datos eficientemente. |
| | **MLflow** | Gestiona todo el ciclo de vida de los modelos de ML, desde el seguimiento de experimentos hasta el registro de modelos |
| | **Bigquery** | Esencial para el análisis de grandes conjuntos de datos de prueba. |
| | **GitLab** | Para el control de versiones de todos los componentes del proyecto, incluidos los scripts de MLflow y los artefactos generados en los ambientes de desarrollo, prueba y producción. |
| | **Terraform** | Para el despliegue de estos componentes utilizando el principio de infraestructura como codigo |

| | |
|---|---|
| **Integración** | Utiliza las capacidades de monitoreo y alertas de GCP, como Cloud Monitoring, para supervisar el rendimiento y la salud de los modelos en producción, incluida la detección de deriva de modelos y el seguimiento de métricas críticas para garantizar su correcto funcionamiento. |

## 3.4 Roadmap de Implementación

Durante nuestra investigación bibliográfica sobre la implementación y operacionalización efectiva de modelos de Machine Learning (ML) en entornos empresariales, hemos identificado un conjunto de fases críticas **[13]** acompañadas de lineamientos esenciales que recomendamos seguir. Estas fases y lineamientos están diseñados para abordar no solo las necesidades tecnológicas sino también operativas y de personal que surgen al adoptar MLOps dentro de una organización. A continuación, presentamos las fases recomendadas junto con sus lineamientos clave:




### Fase 1 – Desarrollo de ML

**Objetivos**:

- Establecer una infraestructura sólida y escalable para soportar el desarrollo y experimentación de ML.
- Clarificar el problema y solución de ML a desarrollar.

**Lineamientos:**

- Identificar y configurar las herramientas y la infraestructura necesarias para el desarrollo, entrenamiento y prueba de modelos de ML
- proporcionando entornos dedicados para garantizar la disponibilidad de recursos y el aislamiento necesario sin afectar la producción.

### Fase 2: Transición hacia Operaciones

**Objetivos**:

- Establecer flujos de trabajo automatizados de CI/CD para el despliegue de modelos.
- Preparar los modelos de ML para un despliegue eficiente y controlado en entornos de producción.

**Lineamientos:**

- Configurar pipelines de integración y entrega continuas para automatizar el despliegue de modelos, asegurando la validación de calidad antes de la puesta en producción.
- Implementar procesos para el despliegue estandarizado de modelos, aprovechando técnicas de conteinerización para la distribución y ejecución de modelos.

- Realizar pruebas exhaustivas para validar el rendimiento y la funcionalidad de los modelos en entornos similares a producción antes de su lanzamiento final.

**Fase 3 – Operaciones**

**Objetivos**:

- Monitorear y gestionar el rendimiento de los modelos en producción para garantizar su efectividad continua.
- Habilitar el aprendizaje continuo a través del reentrenamiento de modelos basado en el rendimiento y los datos operacionales.

**Lineamientos:**

- Establecer sistemas de monitoreo para evaluar continuamente el rendimiento del modelo en producción, identificando deriva del modelo, sesgos y otras métricas relevantes.
- Mantener la gestión eficiente de las tuberías de datos y la seguridad de la plataforma ML, resolviendo proactivamente posibles errores para asegurar un funcionamiento ininterrumpido del sistema de producción.
- Implementar procesos para el reentrenamiento periódico de modelos basados en datos operacionales y retroalimentación del rendimiento, asegurando que los modelos se mantengan actualizados y relevantes.

Al implementar estos objetivos y lineamientos, las organizaciones pueden establecer un ciclo de vida de ML robusto y escalable, garantizando así el éxito sostenible de sus soluciones de ML. La clave es la claridad en el problema a resolver, la selección adecuada de herramientas, la infraestructura, y un enfoque sistemático para la transición y operación de modelos en el mundo real.

# 4 Conclusiones

La conclusión principal de este estudio subraya la importancia crítica de adaptar y optimizar los procesos de Machine Learning Operations para entornos con recursos limitados. A través de la revisión bibliográfica y el análisis realizado, se evidencia que la implementación de MLOps no solo es viable sino esencial para las organizaciones que deseen desarrollar modelos de Machine Learning, permitiéndoles superar desafíos comunes como la falta de personal especializado y la complejidad inherente al ciclo de vida de los proyectos de Machine Learning.

El estudio también ha permitido una inmersión profunda en el ciclo de vida de un proyecto de Machine Learning y la implementación de MLOps, explorando una variedad de tecnologías y marcos de trabajo. Esta revisión ha resaltado la importancia de un enfoque modular y escalable en la implementación de MLOps. Asimismo, ha reforzado la necesidad de colaboración efectiva entre los distintos roles involucrados, subrayando cómo la interacción sinérgica entre estos es fundamental para el éxito de este tipo de proyectos.

Es importante destacar que el éxito en la implementación de MLOps en equipos pequeños depende no solo de la adopción de tecnologías y procesos adecuados, sino también de la formación continua y el desarrollo de una cultura

organizacional que fomente la colaboración, el aprendizaje constante, y la innovación, de forma que la organización mantenga la relevancia y la competitividad en un campo que evoluciona rápidamente como el Machine Learning.

La culminación de nuestra investigación bibliográfica se centra en el desarrollo de un Roadmap de implementación estratégica que esboza las fases críticas y lineamientos esenciales para la adopción efectiva de MLOps en equipos de trabajo pequeños. Este Roadmap identifica el Equipo Mínimo Requerido (con sus roles y responsabilidades), así como el Stack Tecnológico necesario para una implementación coherente y ordenada de MLOps en la organización.

# 5 Referencias Bibliográficas


**[1]** C. Chen, N. Richard Murphy, K. Parisa, D.Sculley, T. Underwood, "Reliable Machine Learning: Applying SRE Principles to ML in Production", O'Reilly Media, 2021.

**[2]** E.Raj, "Engineering MLOps", Packt Publishing, Birmingham, 2021.

**[3]** Y. Haviv & N. Gift, "Implementing MLOps in the Enterprise", O'Reilly Media, 2023. Chapter 2 – "The Stages of MLOps"

**[4]** G. Symeonidis, E. Nerantzis, A. Kazakis, G. Papakostas, "MLOps – Definitions, Tools and Challenges", International Hellenic University, Greece. 2022.

**[5]** D. Kreuzberger, N. Kühl, S. Hirsch. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture". KIT. 2022.

**[6]** V. Tuulos, "Effective Data Science Infrastructure", Manning Publications Co., New York, 2022.

**[7]** Microsoft, "Machine Learning operatins maturity model", https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/mlops-maturity-model , 2024.

**[8]** M. Mary John, H. Holmström Olsson, J. Bosch, "Towards MLOPs: A Framewrok and Maturity Model", Malmö University, Suecia. 2021.

**[9]** DAMA International, "Data Management Body of Knowledge", 2nd Edition, Technics Publications, Basking Ridge – New Jersey, Chapter 2 – Data Handling Ethics

**[10]** DAMA International, "Data Management Body of Knowledge", 2nd Edition, Technics Publications, Basking Ridge – New Jersey, Chapter 7 – Data Security

**[11]** N. Lauchande, "Machine Learning Engineering with MLflow", Packt Publishing, Birmingham, 2021.

**[12]** J. Bhatia, K. Chaudhary, "The definitive guide to Google Vertex AI", Packt Publishing, Birmingham, 2023.

**[13]** D. Tan, A. Leung, D. Colls, "Effective Machine Learning Teams: Best Practices for ML Practicioners", O'Reilly Media, 2024.

# 6 Anexos

## 6.1 Glosario

**Ambiente de Desarrollo:** Es un espacio virtual o físico donde los desarrolladores de software crean, prueban y depuran aplicaciones. Este entorno proporciona las herramientas y recursos necesarios para que los desarrolladores puedan trabajar de manera eficiente y efectiva.

**Ambiente de Testing / QA:** También conocido como entorno de pruebas, es un espacio virtual o físico donde se prueban las aplicaciones de software para verificar su correcto funcionamiento y detectar errores. Este entorno simula las condiciones reales en las que se ejecutará la aplicación, permitiendo a los testers identificar y corregir problemas antes de que la aplicación llegue a los usuarios finales.

**Ambiente de Producción:** También conocido como entorno de producción, es el entorno final donde se ejecuta una aplicación de software para que sea utilizada por los usuarios finales. Este entorno es donde la aplicación se ejecuta en condiciones reales, con los datos reales y bajo la carga de trabajo real.

**Apache Spark:** Es un motor unificado de análisis de datos a gran escala que ofrece una amplia gama de funcionalidades para procesar datos de forma rápida y eficiente. Es una herramienta popular entre los profesionales de datos debido a su flexibilidad, escalabilidad y facilidad de uso.

**Apache Airflow:** Es una plataforma de código abierto para desarrollar, programar y monitorizar flujos de trabajo de data engineering. En español, se traduce a gestión y automatización de tareas relacionadas con el procesamiento de datos.

**Base de Datos SQL:** Es un sistema que almacena y organiza datos de forma estructurada, permitiendo su acceso y manipulación mediante el lenguaje SQL (Structured Query Language).

**Base de Datos NoSQL:** También conocida como base de datos no relacional, es un sistema de almacenamiento de datos que no sigue el modelo de base de datos relacional tradicional. En lugar de utilizar tablas y relaciones entre ellas, las bases de datos NoSQL almacenan los datos en diferentes formatos, como documentos, claves-valor, o grafos.

**CI/CD**: Significa Integración Continua y Entrega Continua (Continuous Integration y Continuous Delivery). Se trata de un conjunto de prácticas automatizadas que agilizan el proceso de desarrollo de software.

**Cloud Storage:** también conocido como **almacenamiento en la nube**, es un modelo de almacenamiento de datos en el que los datos se guardan en servidores remotos ubicados en centros de datos administrados por un proveedor externo. En otras palabras, en lugar de almacenar tus archivos en un disco duro local o en un servidor propio, los alojas en la infraestructura de un proveedor de servicios en la nube.

**DevOps**: Es una cultura, un conjunto de prácticas y una serie de herramientas que unen a los equipos de desarrollo (Dev) y operaciones (Ops) para trabajar de forma colaborativa y eficiente. El objetivo principal es acelerar la entrega de software de calidad a los clientes.

**Docker:** Es una plataforma de código abierto que permite desarrollar, desplegar y ejecutar aplicaciones en contenedores. Éstos son unidades estandarizadas que encapsulan el código fuente de una aplicación junto con todas sus dependencias (bibliotecas, sistema operativo, configuraciones) necesarias para que se ejecute de forma fiable en cualquier entorno.

**Grafana:** es una plataforma de código abierto utilizada para la visualización y análisis de datos, particularmente enfocada en series temporales. En otras palabras, ayuda a transformar tus datos en gráficos y paneles fáciles de entender para que puedas monitorizar y sacar conclusiones sobre el rendimiento de aplicaciones, sistemas, infraestructura, etc.

**GIT:** Es un software de control de versiones diseñado para seguir el historial de cambios en archivos y código fuente. Es especialmente útil para proyectos de desarrollo colaborativo, donde múltiples personas trabajan en el mismo código.

**IDE**: Entorno de Desarrollo Integrado (en inglés, Integrated Development Environment) es una aplicación informática que proporciona un conjunto de herramientas y funcionalidades para facilitar el desarrollo de software.

**Jenkins:** es un servidor de integración continua y entrega continua (CI/CD) de código abierto. Funciona como un automatizador que ejecuta una serie de tareas para garantizar la calidad del código y automatizar el proceso de liberarlo a producción.

**Kubernetes:** Es una plataforma de código abierto para la automatización de la implementación, la gestión y el escalado de aplicaciones en contenedores. Es una herramienta poderosa que ayuda a los equipos a orquestar grandes cantidades de contenedores en diferentes entornos, como centros de datos locales, nubes públicas o nubes híbridas.

**Machine Learning**: Es una rama de la inteligencia artificial que se enfoca en desarrollar técnicas para que las computadoras aprendan sin ser programadas explícitamente. En lugar de seguir instrucciones predefinidas, los algoritmos de aprendizaje automático analizan datos y patrones para mejorar su rendimiento y tomar decisiones por sí mismos.

**MLflow:** Plataforma de código abierto que aborda los desafíos en el despliegue de modelos de aprendizaje automático. Proporciona herramientas para gestionar el ciclo de vida de ML desde la experimentación hasta el despliegue

**MLflow Tracking:** Facilita el registro de experimentos de ML, incluyendo detalles como parámetros y resultados, lo que mejora la reproducibilidad y colaboración al comparar experimentos.

**MLflow Projects:** Este componente permite empaquetar el código de ML en proyectos reutilizables, definiendo dependencias y ejecución, lo que mejora la colaboración y facilita la transferencia entre entornos.

**MLflowModels:** Este componente estandariza cómo empaquetar y desplegar modelos de ML, facilitando su reutilización en distintas plataformas y soportando múltiples bibliotecas, simplificando la transición a producción.

**MLflow Model Registry:** Ofrece un repositorio central para administrar modelos de ML, con funciones como versionamiento y seguimiento de estado, apoyando la gestión colaborativa del ciclo de vida de los modelos.

**Orquestación de Workflow:** se refiere al proceso de organizar, automatizar y coordinar una serie de tareas individuales para lograr un objetivo final. Se asemeja a un director de orquesta que guía a los músicos para crear una sinfonía. En este caso, las tareas son los instrumentos y el objetivo final es la ejecución exitosa del workflow completo.

**Prometheus:** Es un sistema de monitorización de código abierto que se utiliza para recopilar y almacenar métricas de aplicaciones, sistemas e infraestructuras. Es una herramienta popular que ofrece una gran flexibilidad y escalabilidad para monitorizar entornos complejos.

**PyCharm**: Es un entorno de desarrollo integrado (IDE) para el lenguaje de programación Python

**Terraform:** Es una herramienta de Infrastructure as Code (IaC), lo que significa que permite definir y gestionar la infraestructura de tu aplicación o servicio mediante código. En otras palabras, en lugar de configurar manualmente servidores, redes, y otros recursos en la nube o en un centro de datos local, puedes codificar la infraestructura deseada y Terraform la provisiona automáticamente.

**Visual Studio Code (VS Code):** Es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft. Es popular entre los desarrolladores debido a su versatilidad y amplia gama de funcionalidades.

# Tabla de Contenidos