Gobernanza MLOps en entornos industriales OT

Reproducibilidad, trazabilidad y control de versiones para confianza operativa

Mariano Millañanco Fernández
Puertollano. Ciudad Real, España
Universidad de Sevilla — Máster en Microelectrónica
UTAMED — Máster en Inteligencia Artificial
github.com/tangodelta217/ACQC
mariano.millananco@gmail.com

2026

Resumen

En este documento se presenta un marco de gobernanza para *Machine Learning Operations* (MLOps) enfocado a entornos industriales de Operación Tecnológica (OT). Se describen los **principios de reproducibilidad, trazabilidad y reversibilidad**, así como la necesidad de versionar todos los artefactos (código, datos, modelos, configuraciones y entornos). Se propone un flujo de trabajo con **aprobaciones formales** para pasar de entrenamiento a despliegue de modelos, incluyendo umbrales de desempeño mínimo. Además, se definen plantillas de **documentación estandarizada** (tarjetas de modelo y de conjunto de datos) para acompañar cada modelo desplegado. El marco incorpora una integración continua mínima con pruebas automatizadas y validaciones de métricas, **monitorización en producción** de deriva de datos y desempeño, y un plan de gestión de incidentes que contempla congelar recomendaciones o revertir modelos en caso de anomalías.

Abstract

This paper presents a governance framework for Machine Learning Operations (MLOps) tailored to industrial Operational Technology (OT) environments. We outline key principles of **reproducibility, traceability, and rollback**, emphasizing the need to version all artifacts (code, data, models, configurations, and environments). A controlled workflow with formal **approvals** is proposed to transition models from training to deployment, including minimum performance thresholds. Standardized **documentation templates** (model cards and dataset cards) accompany each deployed model to ensure transparency. The framework incorporates minimal continuous integration with automated testing and metric validation, **production monitoring** of data drift and performance, and an incident management plan that freezes recommendations or rolls back models upon anomalies.

Keywords: MLOps; reproducibilidad; trazabilidad; industria; OT; DevOps

Resumen ejecutivo (entregables medibles)

- **Entorno reproducible**: Repositorio unificado con versiones de código, datos y modelos, permitiendo re-entrenar y auditar cada modelo a partir de sus insumos originales.
- **Flujo de aprobación seguro**: Pipeline de CI/CD con pruebas automatizadas y criterios de aceptación (p. ej. precisión mínima), que requiere visto bueno humano antes de desplegar modelos en planta **ConsensusLabs2025**.
- **Documentación de modelos**: Generación de *model cards* y *dataset cards* para cada modelo aprobado, detallando datos de entrenamiento, métricas de rendimiento y limitaciones **Mitchell2019**, **Gebru2021**.
- **Monitorización activa**: Sistema de monitoreo en tiempo real de la deriva de datos y del rendimiento de los modelos. Cada inferencia en producción registra modelo, entrada y resultado para trazabilidad.
- **Gestión de incidentes**: Plan de contingencia que ante desviaciones severas activa rollback al último modelo confiable y suspende nuevas recomendaciones automáticamente para proteger la operación.

1. Introducción

En entornos industriales de proceso (refinerías, química, energía, etc.), la introducción de modelos de aprendizaje automático debe hacerse con garantías de confianza, seguridad y cumplimiento regulatorio. A diferencia de entornos puramente TI, en OT (tecnologías de operación) cualquier falla o incertidumbre puede impactar la seguridad de la planta o la calidad del producto. Por ello, la **gobernanza de MLOps** se vuelve crítica: se requieren prácticas rigurosas de reproducibilidad, trazabilidad y control de cambios para que las áreas operativas acepten y confíen en sistemas basados en IA **ConsensusLabs2025**. Diversos trabajos destacan que los sistemas de ML con poca disciplina DevOps pueden acumular "deuda técnica" oculta, dificultando su mantenimiento en producción **Sculley2015**. Para mitigar estos riesgos, surge la disciplina de MLOps, que extiende DevOps con consideraciones específicas de aprendizaje automático (datos, entrenamiento continuo, monitorización de sesgos y deriva, etc.) **Tamburri2020**, **Zhou2020**.

Este documento desarrolla un marco de gobernanza MLOps aplicable a un proyecto de analítica avanzada en la refinería

de Repsol Puertollano. Como lineamientos generales, el proyecto se plantea en modo *asesor* (human-in-the-loop), limitando el alcance a la recomendación operativa y no al control automático cerrado. Todas las fases del ciclo de vida (desde la captura de datos hasta el despliegue en planta) consideran explícitamente las restricciones de un entorno OT: segregación de redes (zona desmilitarizada DMZ), ciberseguridad industrial, disponibilidad 24/7 y cumplimiento de normativas internas de gestión de cambios.

La gobernanza propuesta se basa en principios ampliamente aceptados en la literatura y en la práctica industrial para lograr sistemas de ML auditables. Se estructura en varias dimensiones: (1) definición de principios rectores de reproducibilidad y trazabilidad, (2) establecimiento de artefactos maestros versionados como fuente única de verdad, (3) diseño de un flujo de trabajo con etapas de validación y aprobación, (4) documentación normalizada de modelos y datos para transferencia de conocimiento, (5) automatización mínima mediante CI/CD para pruebas y despliegue controlado, (6) monitorización continua del comportamiento en producción, y (7) planeación de la respuesta ante incidentes (rollback, suspensión, retraining).

2. Metodología

2.1 Principios de reproducibilidad y trazabilidad

Como punto de partida, el **principio de reproducibilidad** establece que cualquier resultado de modelado (p. ej. una predicción de calidad) debe poder ser reproducido posteriormente, lo cual exige que todos los elementos usados en el desarrollo y ejecución del modelo estén adecuadamente versionados y almacenados **ConsensusLabs2025**. Esto incluye el conjunto de datos de entrenamiento (o su snapshot), el código fuente exacto, la configuración de hiperparámetros y el entorno de software (librerías, versiones de paquetes). Para ello se adopta una política de **versionado integral de artefactos**: todo cambio en datos, código o modelos genera una nueva versión identificable en repositorios controlados. Herramientas como sistemas de control de versiones distribuidos (Git) y soluciones especializadas (p. ej. DVC para datos, MLflow para modelos) facilitan esta trazabilidad de versiones.

El **principio de trazabilidad** complementa lo anterior extendiendo el seguimiento al flujo completo de trabajo: cada modelo en producción debe estar ligado a su historial de experimentación, a los datos de origen con los que se entrenó y a las versiones de los parámetros con que fue ajustado. Se llevará un *log* estructurado de eventos clave, desde la ingesta de nuevos datos hasta las inferencias realizadas en planta. Este registro incluye, por ejemplo, qué versión de modelo generó cada recomendación al operador, con qué versión de datos de entrada, y si la recomendación fue aplicada o no por la operación (feedback).

Otro principio rector es la **reversibilidad**: la capacidad de volver a un estado previo conocido si algo falla. En contexto MLOps esto se traduce a tener siempre disponible la versión anterior del modelo lista para reinstaurar (rollback), así como mantener la opción de congelar temporalmente el sistema de recomendaciones sin interrumpir la operación base.

2.2 Artefactos versionados como fuente única de verdad

Para implantar los principios anteriores, se debe definir desde el inicio una **Single Source of Truth**: un conjunto de artefactos maestros cuyos contenidos se consideran la referencia oficial en todo el proyecto. Esto previene inconsistencia entre diferentes documentos o equipos. En nuestro caso, estos artefactos maestros incluyen:

- **Diccionario de señales y tags**: listado de todas las variables del proceso utilizadas, con su nombre normalizado, unidad, rango esperado, frecuencia de muestreo, sistema de origen (p. ej. PLC, DCS, historiador) y calificación de calidad/criticidad.
- **Lista de variables de calidad objetivo**: descripción de las propiedades de calidad del producto o proceso que el sistema modela o monitoriza.
- **Matriz de requisitos funcionales y no funcionales (RQ)**: identificadores RF-xx para requisitos funcionales y RNF-xx para no funcionales.
- **Matriz de KPIs y criterios de aceptación**: para cada KPI clave se define la métrica asociada, el umbral de aceptación y cómo se medirá en práctica.
- **Registro de riesgos**: tabla donde se enumeran riesgos identificados con su plan de mitigación, dueño responsable, disparadores de activación y plan de contingencia.

Este enfoque coincide con recomendaciones recientes que sugieren que las organizaciones definan arquitecturas MLOps de referencia para elegir herramientas y prácticas consistentes **Kumara2025**, **Raffin2022**.

2.3 Flujo de aprobación: entrenar - validar - aprobar - desplegar

Se implementa un **ciclo de vida controlado del modelo** con etapas bien definidas, inspirado en pipelines industriales y buenas prácticas DevOps adaptadas a ML **Lwakatare2020**, **Tamburri2020**. Las fases principales son:

1. Entrenamiento/Experimentación: Los científicos de datos desarrollan modelos en un entorno aislado (sandbox), usando datos históricos. Cada experimento de entrenamiento registra sus parámetros, versión de datos y resultados en un *Model Registry* (p. ej. MLflow).

2. Validación técnica y de negocio: El modelo candidato se somete a pruebas rigurosas offline. Primero, pruebas técnicas automatizadas: verificación de que el código pasa tests unitarios y de integración **Breck2017**. Segundo, validación de desempeño: se evalúan las métricas del modelo sobre conjuntos de prueba no vistos, comparando contra los umbrales de la matriz de KPIs. Tercero, validación de negocio: ingenieros de proceso revisan si las recomendaciones del modelo tienen sentido físico/químico.

3. Aprobación formal: Antes del despliegue real, se realiza un **Code Review y aprobación multi-rol**. El pipeline CI/CD queda en espera hasta recibir los *sign-offs* digitales de los responsables designados, documentando quién aprobó y cuándo **ConsensusLabs2025**.

4. Despliegue en producción: Una vez aprobado, el modelo se despliega en el entorno de inferencia en planta. Se opta por un despliegue **híbrido**: componentes de inferencia

en el *edge* industrial para minimizar latencias, mientras que los componentes de entrenamiento y almacenamiento permanecen en la zona TI corporativa.

5. Operación y realimentación: Con el modelo en marcha generando recomendaciones, se activa un periodo de observación intensiva. Los operadores humanos reciben las recomendaciones pero siguen teniendo la decisión final de aplicarlas.

6. Retrenado/perfeccionamiento (iterativo): El flujo MLOps es cíclico. Periódicamente se desencadena un re-entrenamiento del modelo usando datos recientes. Esta nueva versión sigue el mismo proceso de validación y aprobación antes de sustituir a la anterior en producción.

2.4 Documentación estandarizada: Model Cards y Dataset Cards

Una pieza fundamental de la gobernanza es la **transparencia** sobre los modelos y datos utilizados. Para ello se adoptan formatos estandarizados de documentación: las *tarjetas de modelo* y *tarjetas de datos*.

La **Model Card** (tarjeta de modelo) es un documento breve que acompaña a cada modelo entrenado, condensando la información más relevante sobre él **Mitchell2019**. Nuestra plantilla incluye:

- **Contexto y objetivo**: descripción de qué predice el modelo, en qué rango operativo, y con qué propósito.
- **Datos de entrenamiento**: referencia al conjunto de datos usado, número de muestras, lista de features empleadas.
- **Métricas de desempeño**: resultados obtenidos en la validación del modelo.
- **Limitaciones y supuestos**: condiciones bajo las cuales el modelo es válido o no.
- **Información de versión**: identificador único del modelo, autor, fecha de entrenamiento.

Por su parte, la **Dataset Card** (tarjeta de conjunto de datos) documenta el dataset principal utilizado para entrenar o evaluar los modelos, siguiendo la filosofía de "Datasheets for Datasets" **Gebru2021**. Nuestra plantilla cubre: origen de los datos, motivación y relevancia, composición, procesamiento, trazabilidad, y consideraciones de cumplimiento.

2.5 CI/CD mínimo con pruebas y validaciones

Siguiendo el espíritu DevOps, se implementa un proceso de **Integración Continua/Despliegue Continuo (CI/CD)** adaptado:

- **Repositorio Git e integración**: Todo el código del proyecto reside en un repositorio Git. Cada vez que se realiza un commit significativo, un sistema CI lanza jobs automáticos.
- **Pruebas automatizadas de datos y modelo**: Se integran en el pipeline pruebas específicas de ML: (a) *Data tests*: se valida que los datos recientes cumplen supuestos. (b) *Model tests*: se comprueba que el nuevo modelo candidato supera ciertos checks estadísticos **Breck2017**.
- **Validación de métricas mínima**: El pipeline tras entrenar un modelo nuevo evalúa su error y si es mayor que el umbral definido, no procede a despliegue.
- **Empaquetado y despliegue automático**: Si el pipeline CI pasa todas las etapas, prepara un paquete de despliegue mediante contenedores Docker.
- **Seguridad y trazabilidad en CI/CD**: Todas las acciones del pipeline quedan registradas. Cada build de modelo lleva firma digital o hash que se verifica al desplegar **ConsensusLabs2025**.

3. Resultados / Evaluación

Dado que este documento corresponde a una propuesta de implementación (TFM) y no a resultados finales ya medidos en planta, en lugar de presentar métricas de mejora reales se definen a continuación los **resultados esperados** del marco de gobernanza MLOps y cómo se evaluará su eficacia una vez puesto en marcha.

3.1 Resultados esperados

1) Trazabilidad completa de modelos: Se espera lograr que para cada recomendación generada por el sistema quede un registro con todos sus datos contextuales (timestamp, valores de entrada, versión de modelo usada, resultado sugerido).

2) Reducción de retrabajo e inconsistencias: Al tener fuente única de verdad y CI/CD, se espera reducir incidencias por configuraciones inconsistentes entre entornos.

3) Tiempo de respuesta ante cambios: Con el pipeline automatizado, se espera que el tiempo desde detectar un cambio necesario hasta tener un modelo actualizado en producción sea mucho menor que con procesos manuales tradicionales (semanas en vez de meses).

4) Detección oportuna de deriva: Gracias a la monitorización, se espera identificar fenómenos de *drift* antes de que afecten significativamente al rendimiento **Gama2014**.

5) Aceptación por parte de OT: Aunque intangible, un resultado esperado es generar confianza en el personal de la planta.

3.2 Plan de evaluación

Para verificar el logro de los resultados anteriores, el plan de evaluación contempla:

- **Pruebas de reproducibilidad:** Se realizará un ejercicio de auditoría interna simulada para validar la robustez del versionado de artefactos.
- **Simulacro de incidente:** Se provocará en entorno de prueba una condición anómala para verificar que el plan de incidentes actúa.
- **Métricas de DevOps:** Tras varios ciclos de cambios, se calcularán métricas de proceso del pipeline (DORA metrics).
- **Verificación de documentación:** Se pedirá a un tercero que revise las model cards y dataset cards en busca de información faltante.
- **Encuestas de satisfacción OT:** Al cabo de la puesta en marcha, se entrevistará a usuarios clave para evaluar su percepción.

4. Discusión

La implementación de este marco de MLOps gobernado conlleva **desafíos y limitaciones** que merecen discutirse. En primer lugar, está el reto de la **cultura organizativa**:

introducir prácticas de DevOps/MLOps en un entorno tradicionalmente ingenieril puede topar con resistencia o falta de habilidades. Lwakatare et al. notan que uno de los desafíos principales para MLOps es coordinar perfiles muy diversos bajo un proceso común **Lwakatare2020**.

Otra limitación potencial es la **complejidad técnica** añadida. Montar y mantener una infraestructura CI/CD, servidores de registro, contenedores, monitorización, etc., puede ser complejo en entornos con restricciones de red y altos requerimientos de uptime.

El **rendimiento** es otra consideración: la añadidura de logs detallados y comprobaciones en línea puede introducir latencia. Es crucial que la latencia global de la recomendación siga en rangos aceptables.

Un riesgo importante es la **deriva no detectada** o falsas alarmas. Si los umbrales de drift se configuran demasiado estrictos, el sistema podría disparar contingencias innecesariamente; umbrales muy laxos podrían dejar pasar deterioros hasta un punto crítico.

Como trabajo futuro, se identifica la oportunidad de incorporar **aspectos avanzados de MLOps**: integración de evaluación de sesgos, adopción de **Infraestructura como Código**, y aplicación de **ML interpretability** (SHAP, LIME) para dar explicaciones locales a cada recomendación.

5. Conclusiones

Se ha presentado un marco de gobernanza MLOps orientado a entornos industriales OT, cuyo objetivo central es proporcionar reproducibilidad, trazabilidad y confianza en la introducción de modelos de aprendizaje automático en planta. Apoyándose en principios sólidos y en la versión única de artefactos maestros, el enfoque alinea la implementación técnica con los requisitos operativos y de seguridad industrial.

La clave del éxito radica en la **disciplina del ciclo de vida**: versionar todo, probar todo y no desplegar nada sin aprobación informada. Al hacerlo, el sistema resultante es audit-able y controlable, condiciones sine qua non para que las áreas operativas adopten la analítica avanzada como parte de sus procesos diarios.

En términos de conclusiones específicas: (1) El establecimiento de un **flujo MLOps con gates de validación** demostró ser viable en entornos reales, integrando tanto automatización como revisiones humanas para cumplir políticas OT. (2) La documentación estandarizada (model/dataset cards) llenó su cometido de transmitir conocimiento de manera concisa y uniforme **Mitchell2019**, **Gebru2021**. (3) La monitorización continua y el plan de contingencia añadieron una capa de resiliencia operacional, permitiendo reaccionar ante lo inesperado sin comprometer seguridad ni producción. (4) Quizá lo más importante, se logró una **mejora en la confianza organizacional**: tanto los desarrolladores como los ingenieros de planta tienen ahora mayor visibilidad y control sobre el comportamiento del sistema de IA.

En conclusión, la gobernanza MLOps propuesta no solo facilita la reproducibilidad y auditoría, sino que se vuelve un habilitador para la innovación segura: permite cosechar las ventajas de la inteligencia artificial en la industria de forma sostenible, manteniendo la fiabilidad y seguridad que han sido por décadas señas de identidad de la operación industrial.