

Bogotá D.C., 14 de julio de 2025

REPORTE FASE 4

TRABAJO FINAL DE MAESTRÍA “PRONÓSTICO DE LA PRECIPITACIÓN ACUMULADA EN COLOMBIA, UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO”

NOMBRE: JUAN DIEGO MOGOLLÓN OVIEDO	
OBJETIVO: Desarrollar un modelo para el pronóstico de la precipitación acumulada en Colombia, utilizando técnicas de aprendizaje automático.	
OBJETIVOS ESPECÍFICOS: <ul style="list-style-type: none">• Seleccionar un conjunto de datos abiertos de precipitación acumulada en Colombia, verificando su acceso y calidad mediante algoritmos de análisis de datos.• Implementar un proceso para la extracción, transformación y carga de los datos, aplicando técnicas de ingeniería de características.• Evaluar el desempeño de diferentes algoritmos de aprendizaje automático para el pronóstico de precipitación acumulada, utilizando los datos procesados y métricas de evaluación empleadas en predicción climática.• Diseñar una arquitectura para la implementación del modelo en un entorno de prueba, aplicando prácticas y herramientas de operaciones de aprendizaje automático.	
PLAZO DE EJECUCIÓN: 16 semanas	
INICIO: 31 de marzo de 2025	FINALIZACIÓN: 26 de julio de 2025

Con el objetivo de dar cumplimiento al cronograma estipulado en la “Propuesta de Trabajo Final de Maestría” en el periodo 09 de junio de 2025 al 20 de julio de 2025, se presenta a continuación el reporte de la fase 4:

RESULTADO OBTENIDO: <p>Se diseñó una arquitectura de implementación del modelo en un entorno de prueba, aplicando prácticas y herramientas de operaciones de aprendizaje automático.</p> <p>Para esto, se realizó una búsqueda de bibliografía académica y servicios web que permitió definir las prácticas (principios, componentes) y herramientas de software.</p> <p>Los siguientes principios guían el diseño de la arquitectura para asegurar un ciclo de vida de ML automatizado, colaborativo y de alta calidad:</p> <ul style="list-style-type: none">• P1: Automatización (CI/CD): Automatizar las etapas de construcción, prueba, entrega y despliegue. Proporciona retroalimentación sobre posibles fallos en el proceso, lo que mejora la eficiencia del desarrollo y contribuye a una mayor productividad.• P2: Versionamiento: Versionar no solo el código, sino también los datos y los modelos para garantizar la trazabilidad completa de los resultados.

- **P3: Reproducibilidad:** Asegurar que un experimento de ML pueda ser replicado, obteniendo los mismos resultados si se utilizan los mismos datos, código y parámetros.
- **P4: Colaboración:** Fomentar la comunicación y el trabajo colaborativo entre los diferentes roles involucrados (científicos de datos, ingenieros de ML, operaciones).
- **P5: Entrenamiento Continuo:** Reentrenar y validar el modelo de forma periódica o ante cambios en los datos para mantener su rendimiento y relevancia.
- **P6: Monitoreo:** Supervisar de forma proactiva el rendimiento del modelo en producción, así como la infraestructura subyacente, para detectar anomalías o degradación.
- **P7: Retroalimentación:** Establecer ciclos de retroalimentación que permitan utilizar los resultados del monitoreo para mejorar el modelo y el sistema.

A continuación se detallan los componentes técnicos utilizados, los principios MLOps que soportan y las herramientas recomendadas.

Componente y Descripción	Principios MLOps Asociados	Herramientas Recomendadas
C1: Componente CI/CD Automatiza la construcción, prueba y despliegue de los pipelines.	P1 (Automatización), P5 (Entrenamiento Continuo), P7 (Retroalimentación)	Jenkins, GitLab CI/CD, GitHub Actions
C2: Repositorio de Código Fuente Almacena y versiona todo el código (pipelines, pruebas, configuración).	P2 (Versionamiento), P4 (Colaboración)	Git, GitHub, GitLab, Bitbucket
C3: Orquestación de Flujos de Trabajo Define, programa y ejecuta los flujos de trabajo de ML mediante grafos dirigidos acíclicos (DAGs).	P3 (Reproducibilidad), P5 (Entrenamiento Continuo)	Apache Airflow, Kubeflow Pipelines, Argo Workflows
C4: Almacén de Características (Feature Store) Sistema centralizado para almacenar, gestionar y servir características de ML.	P3 (Reproducibilidad), P4 (Colaboración)	Feast, Tecton, Influxdata
C5: Infraestructura de Entrenamiento Proporciona los recursos computacionales (CPUs, RAM y GPUs) para entrenar modelos.	P5 (Entrenamiento Continuo)	Máquinas locales (no escalable) o Computación en la nube: AWS, GCP, Azure
C6: Registro de Modelos Base de datos para almacenar, versionar y gestionar los modelos entrenados.	P2 (Versionamiento), P3 (Reproducibilidad)	MLflow Registry, Neptune.ai, Weights & Biases
C7: Almacén de Metadatos ML Registra información sobre cada ejecución del pipeline (métricas, parámetros, etc.).	P2 (Versionamiento), P3 (Reproducibilidad)	MLflow Tracking, Neptune.ai, Weights & Biases
C8: Componente de Servicio del Modelo Despliega modelos como servicios accesibles (ej. API REST).	P1 (Automatización)	Seldon Core, KServe, TensorFlow Serving, FastAPI
C9: Componente de Monitoreo Realiza el seguimiento continuo del	P6 (Monitoreo), P7 (Retroalimentación)	MLflow, Prometheus, Grafana, Evidently AI



rendimiento del modelo en
producción.

A partir de estos conceptos, se diseñó la arquitectura utilizando Canva para diagramar y representar visualmente la integración de los distintos módulos del sistema. El resultado se muestra en la siguiente figura:

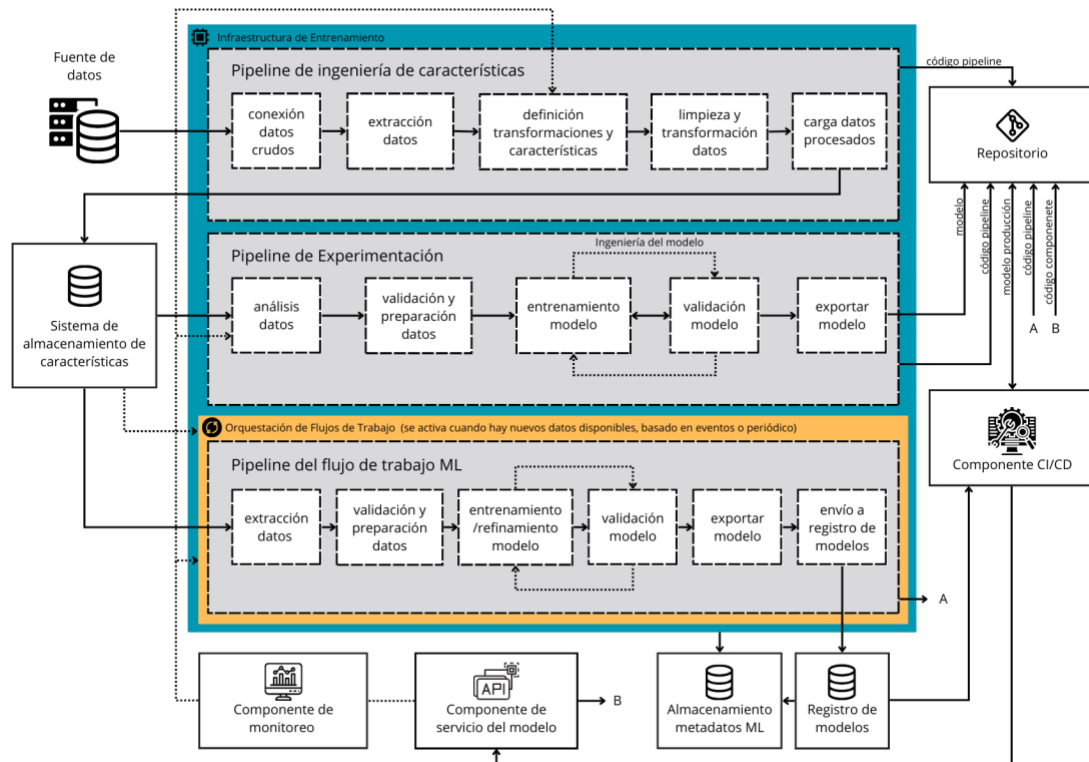


Figura 1. Diagrama de la arquitectura de implementación del modelo.

ACTIVIDADES REALIZADAS:

1. Se creó el diagrama "arquitectura.pdf" con el diseño de la arquitectura de implementación del modelo.

ENLACE: [Repositorio - GitHub](#)

Jorge Eliecer Camargo Mendoza
Director

Juan Diego Mogollón Oviedo
Estudiante