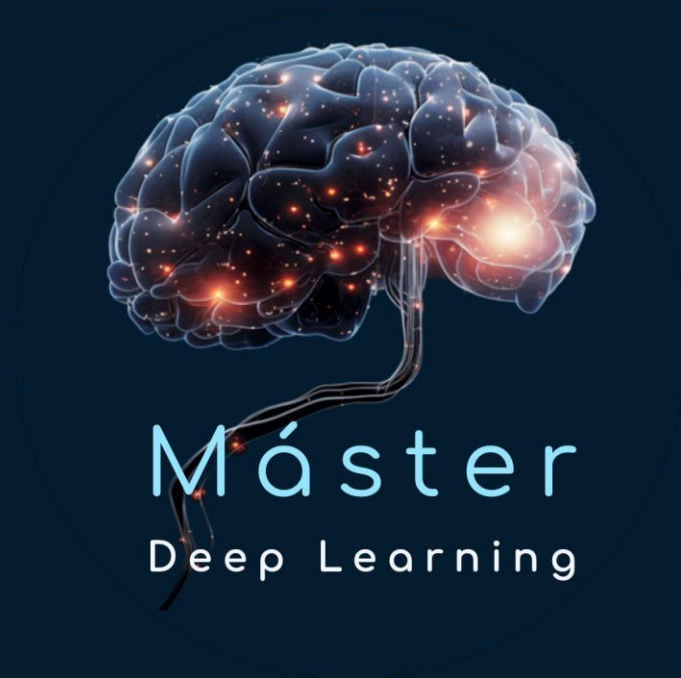


# MLOps

Tema 1

## Introducción a MLOps



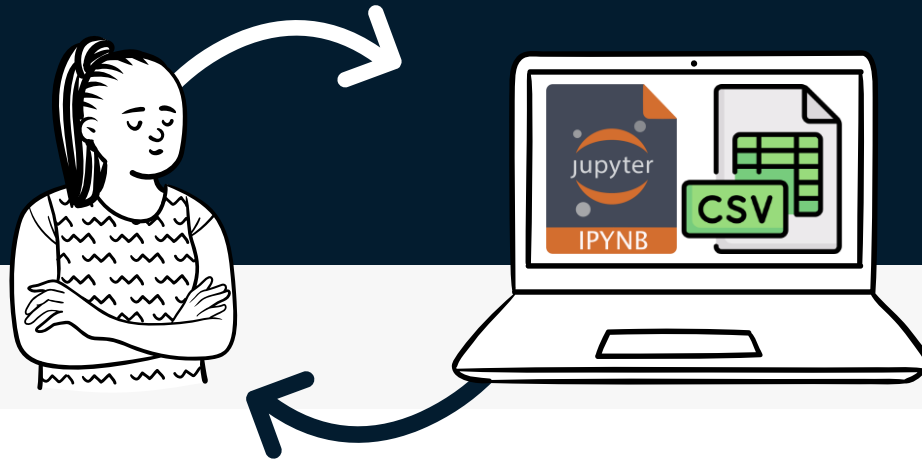
POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

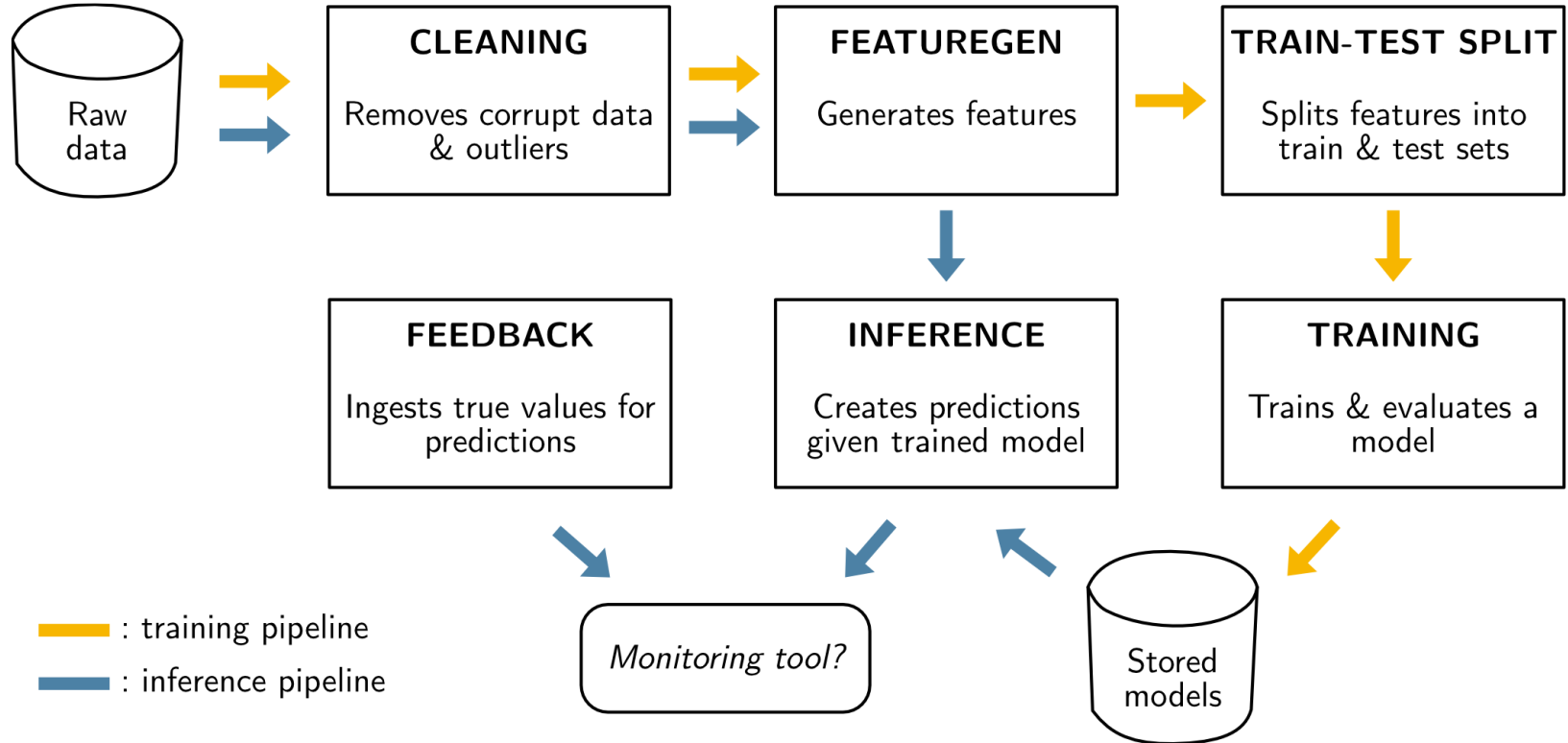


Máster  
Deep Learning

Una científica de datos predice las ventas diarias de un comercio usando el histórico de ventas



# Workflow local de ML



**Notebook  
único**

**Sin control  
de versiones**

ML  
code

**Actualizaciones manuales**

**Sin colaboración ni escalabilidad**

Un grupo de ingenier@s despliega el predictor de ventas  
para ajustar automáticamente la página del comercio

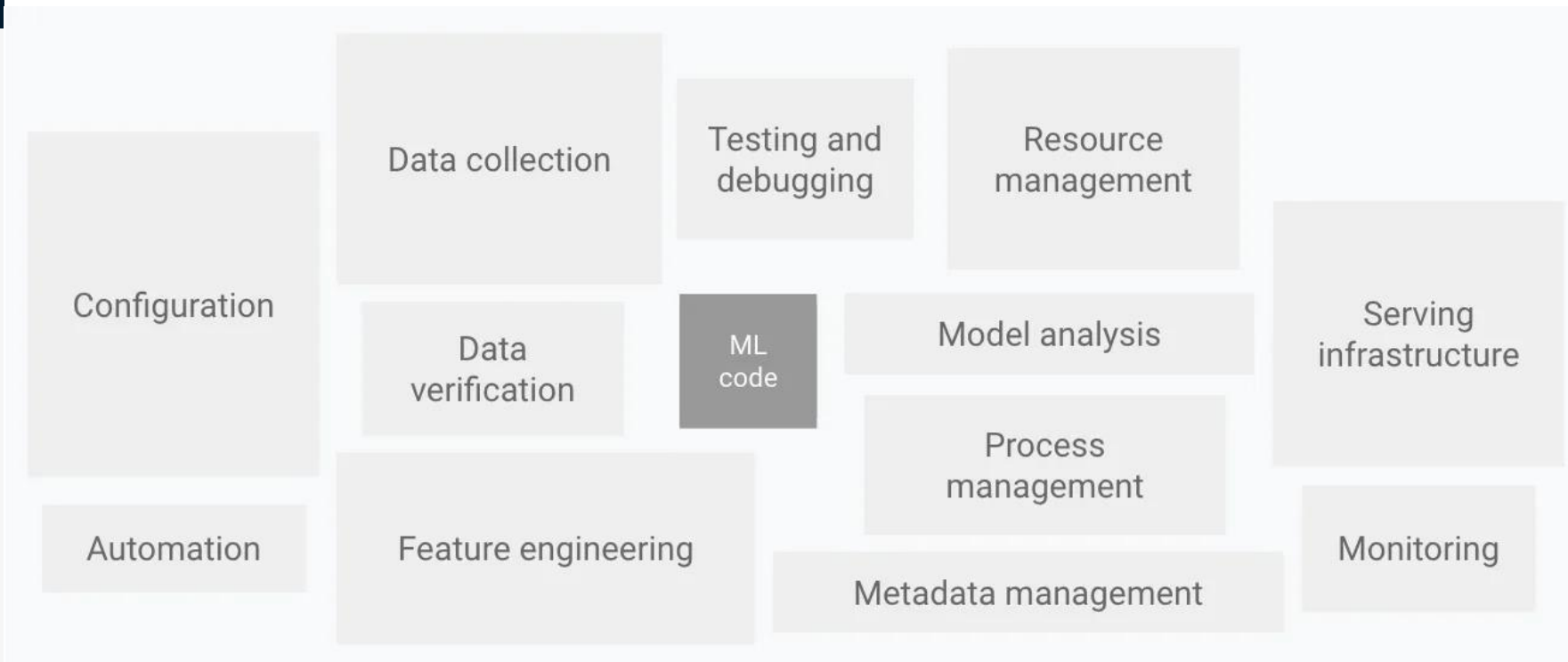


# ML en producción: realidad



1. Elegir una métrica para optimizar
2. Recopilar datos
3. Entrenar el modelo
4. Darse cuenta de que muchas etiquetas están mal → volver a etiquetar los datos
5. Entrenar el modelo
6. El modelo tiene un mal desempeño en una clase → recopilar más datos para esa clase
7. Entrenar el modelo
8. El modelo tiene un mal desempeño en los datos más recientes → recopilar datos más recientes
9. Entrenar el modelo
10. Desplegar el modelo
11. Soñar con \$\$\$
12. Despertarse porque el modelo muestra sesgo contra un grupo → revertir a la versión anterior
13. Conseguir más datos, entrenar más, hacer más pruebas
14. Desplegar el modelo
- 15. Rezar**
16. El modelo funciona bien pero los ingresos están disminuyendo
- 17. Llorar**
18. Elegir una métrica diferente
19. Empezar de nuevo

# Más allá del notebook

















- ▶ Despliegue masivo de modelos en línea que deben funcionar correctamente sin intervención humana
  - ▶ Muchas dependencias
  - ▶ Diferentes lenguajes y equipos
  - ▶ Data scientists no son Software Engineers
- ▶ Los sistemas **ML son muy complejos:**
  - ▶ Más automatización e infraestructura
  - ▶ Para tener menos conocimiento e ingeniería



# Retos

## SOFTWARE 1.0 - WRITE CODE

Design	Version Code	Code & Collaborate	Deploy to Infra	CI/CD	Production Monitoring
 Figma  Sketch	 GitHub  GitLab	 Jira  VS Code	 HashiCorp  puppet	 circleci  Jenkins	 PagerDuty  DATADOG

## SOFTWARE 2.0 - TRAIN MODELS

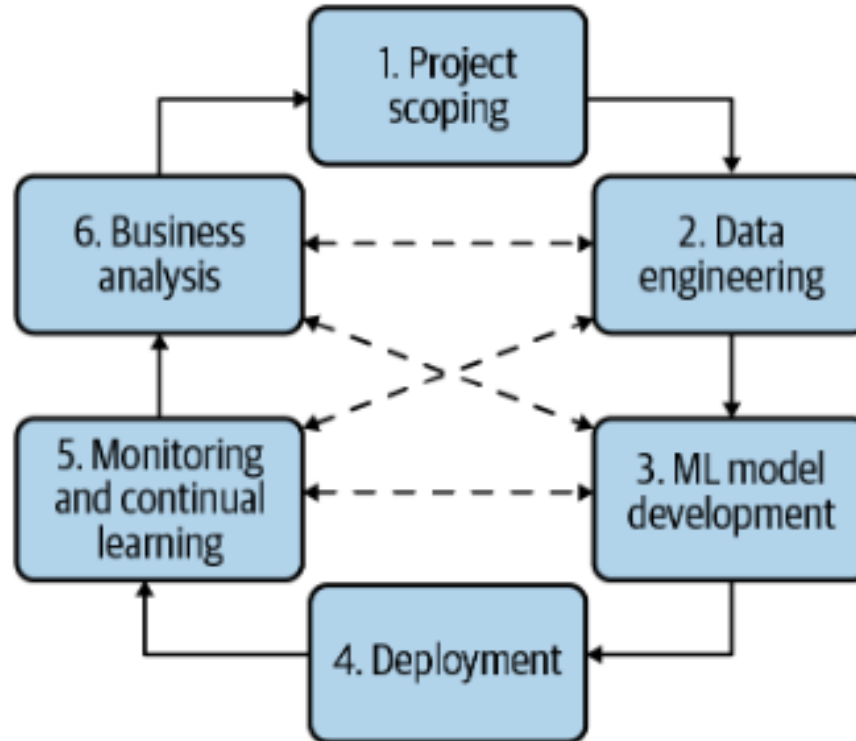
Prep & Visualize Data	Version Data & Models	Experiment Tracking	Manage Model Pipeline	Model CI/CD	Production monitoring
Custom apps Notebooks	Files in S3	Text files Screenshots	Text files	Custom scripts	Nothing

# Surgimiento de MLOps

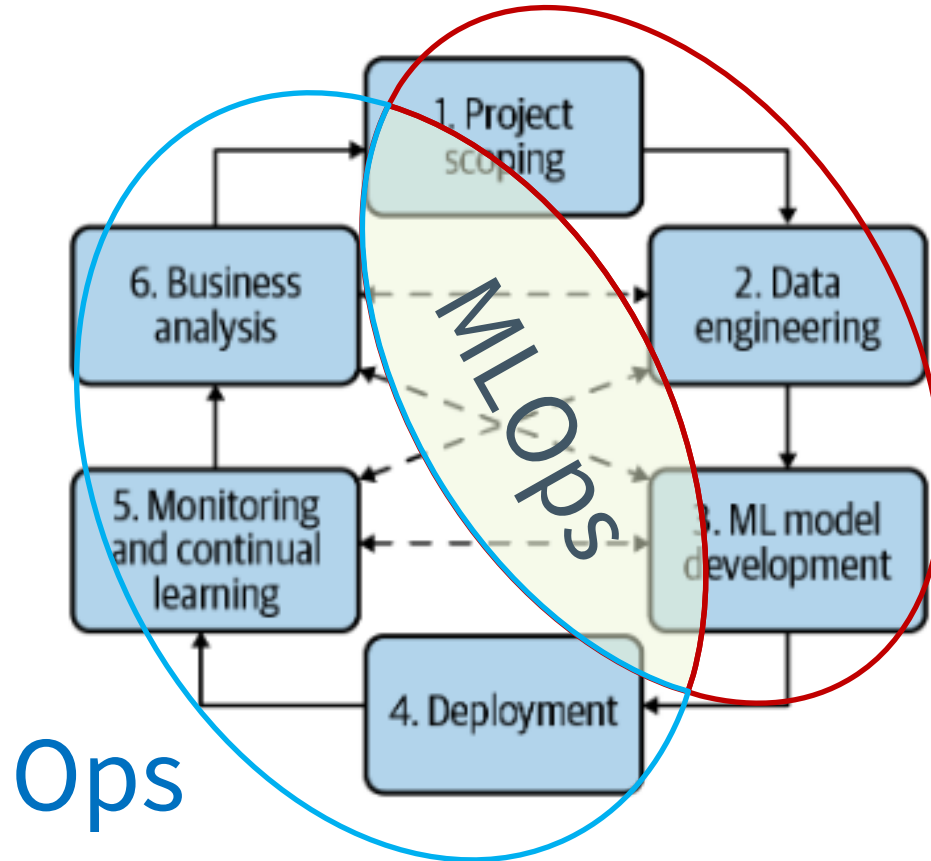
# MLOps

Automatización del ciclo de vida  
de los modelos de ML en producción

# Machine Learning + Operaciones

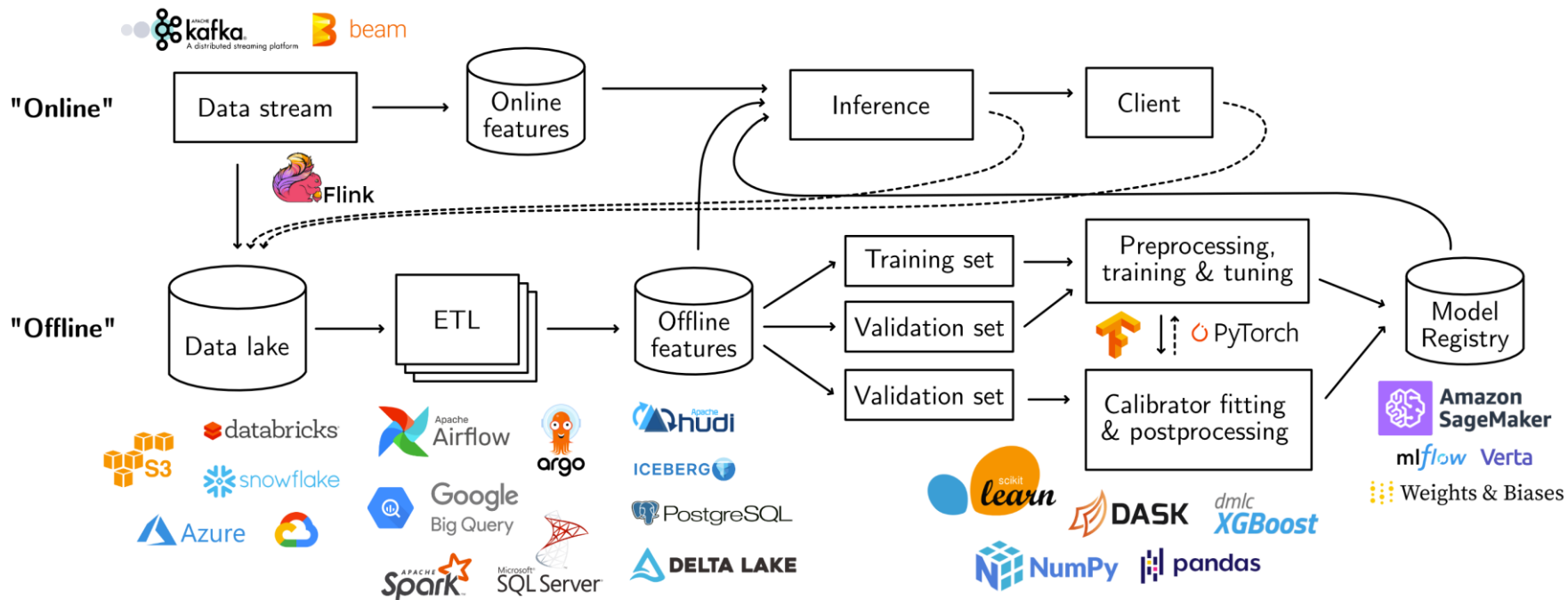


# Machine Learning + Operaciones



ML

# MLOps: Herramientas

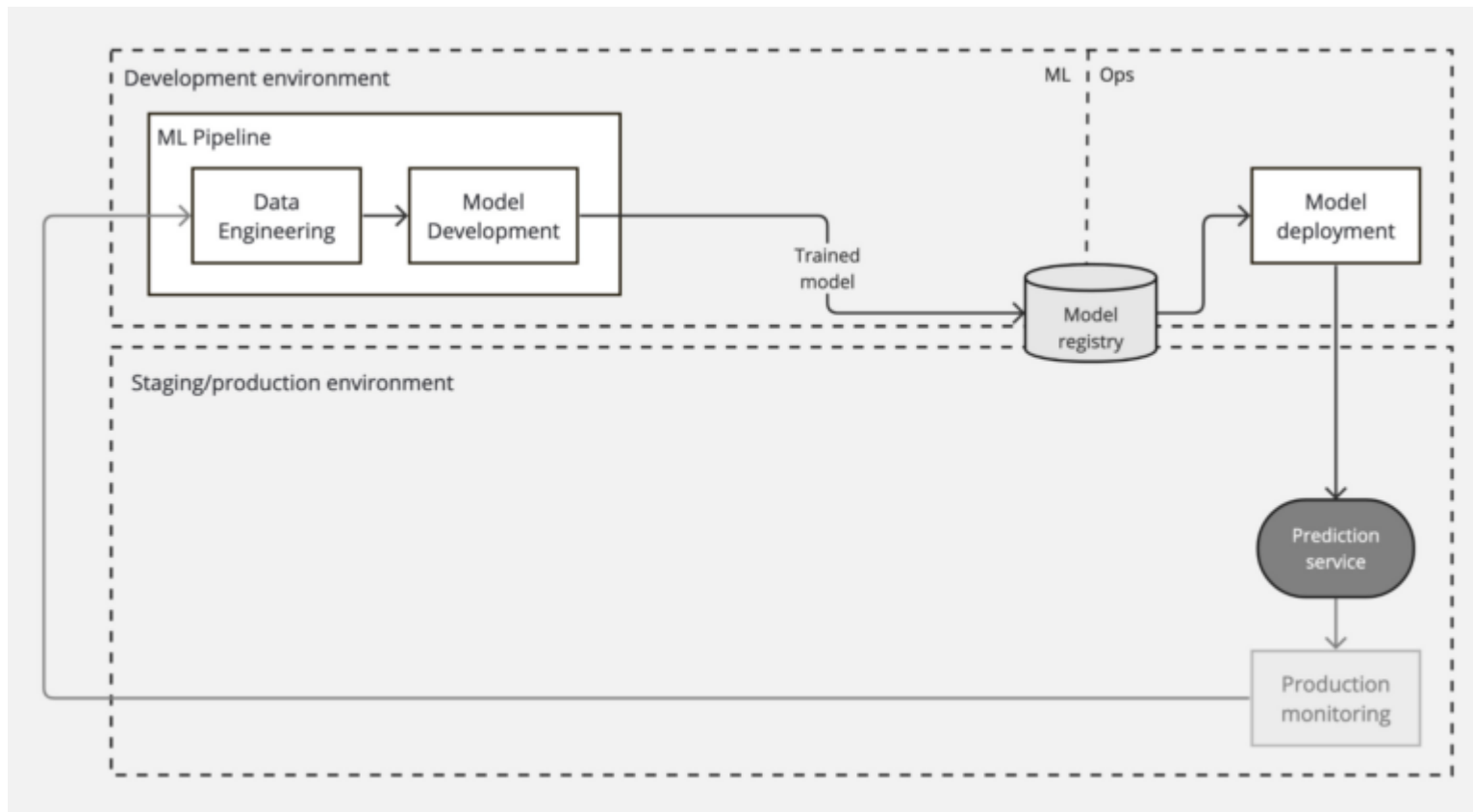


# MLOps Workflows

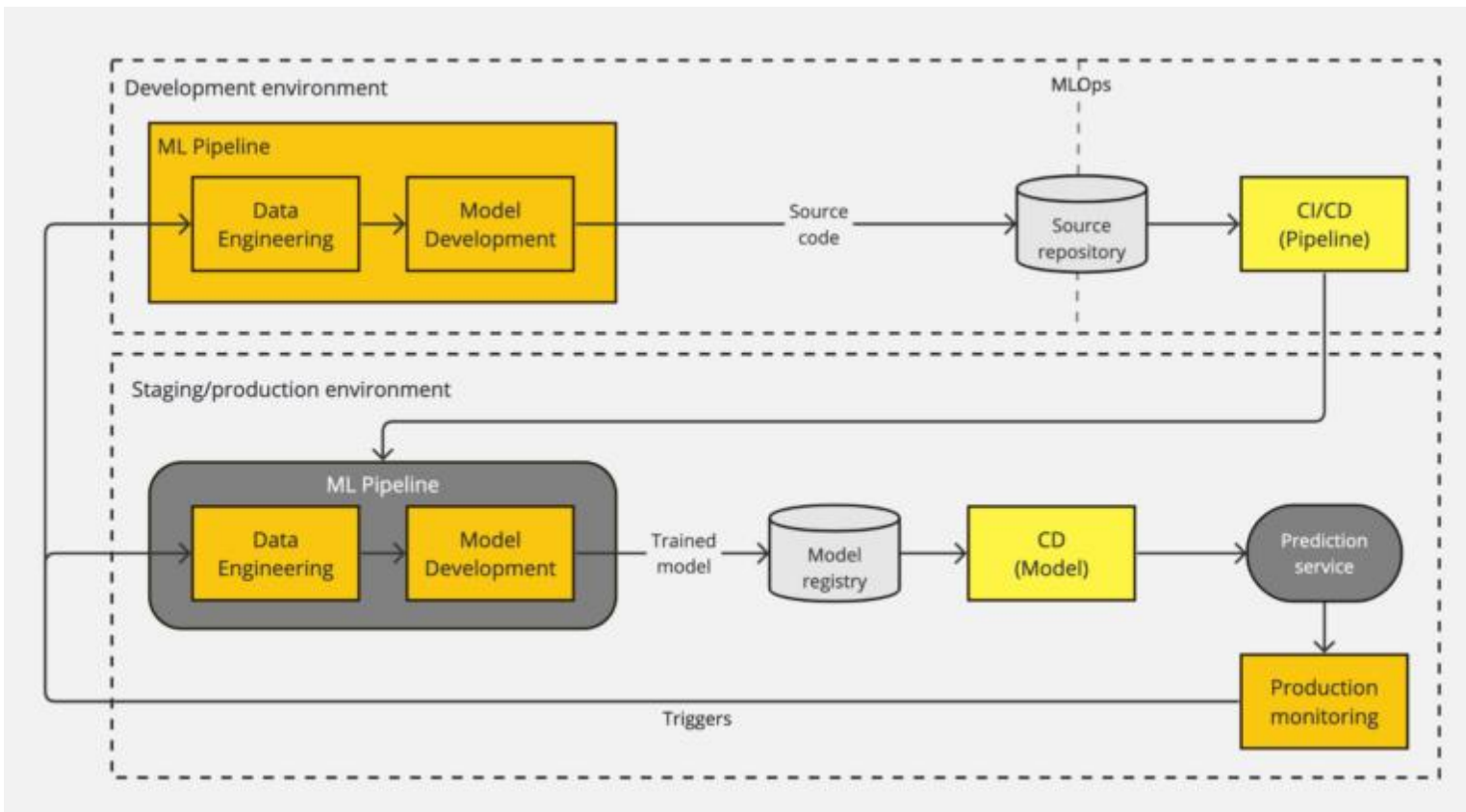
## Diferentes niveles de automatización

*(reflejan la velocidad de entrenar nuevos modelos  
por tener nuevos datos o implementaciones)*

# MLOps Manual

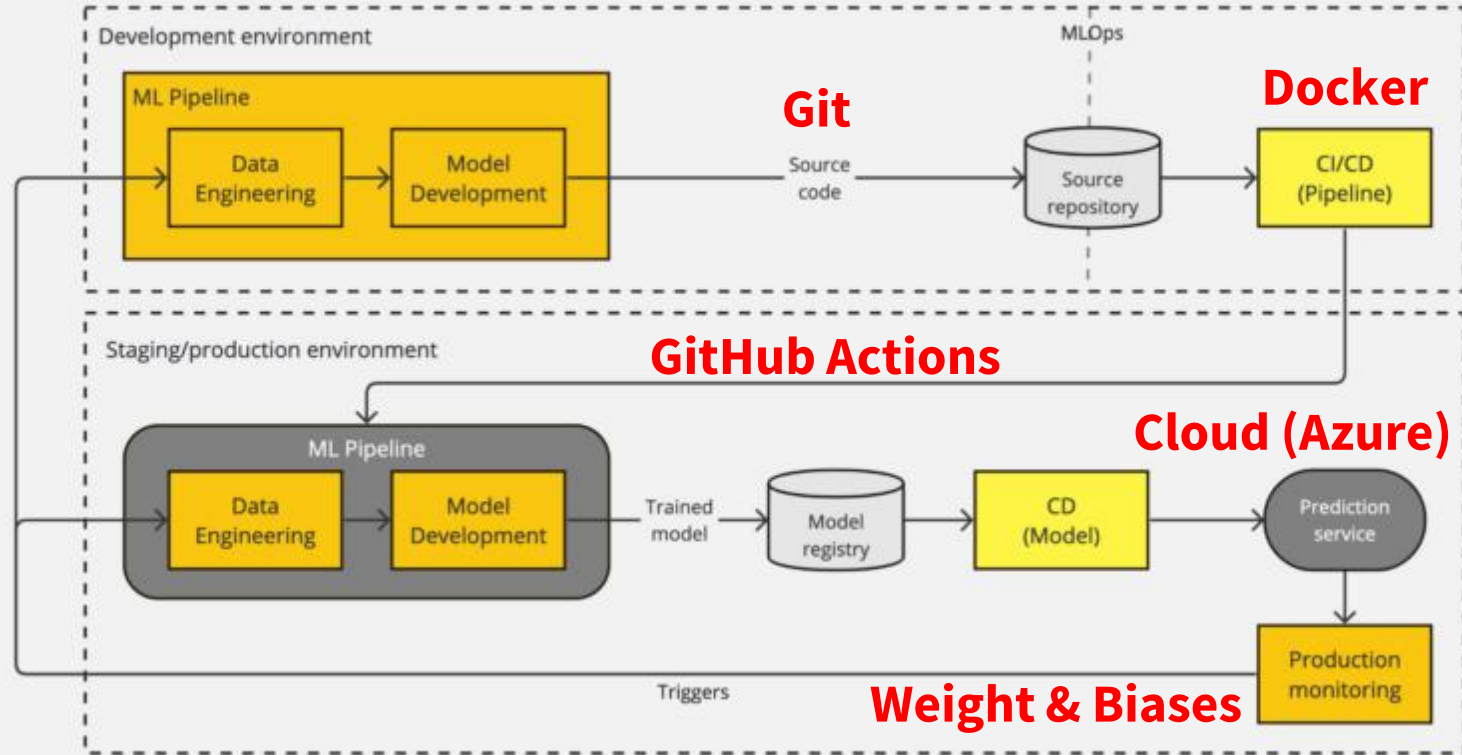


# MLOps Automatizado con CI/CD





# MLOps Automatizado con CI/CD



# Buenas prácticas MLOps

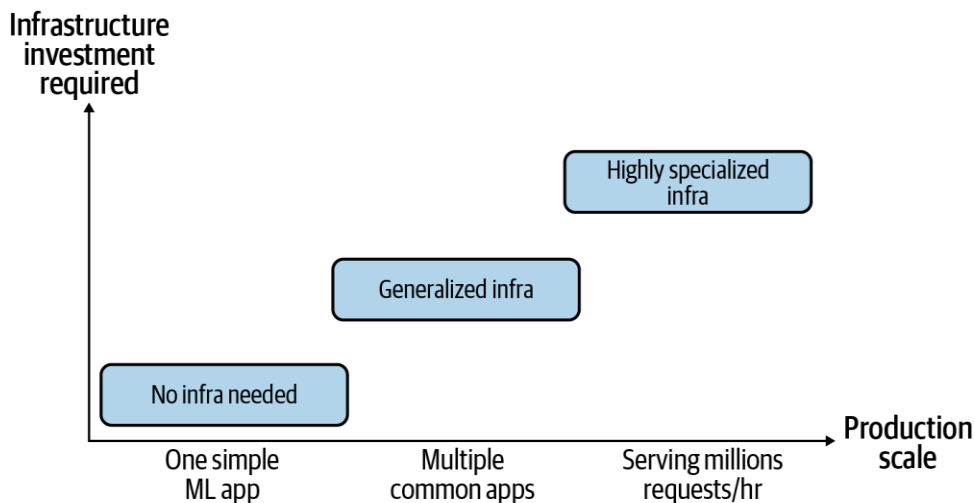
- Versionado de experimentos y datasets
- Entender cuando modelos reentrenados son mejores que versiones anteriores
- Lanzar a producción los mejores modelos
- Monitorear periódicamente el rendimiento del modelo para evitar su degradado en producción

# Infraestructura para MLOps

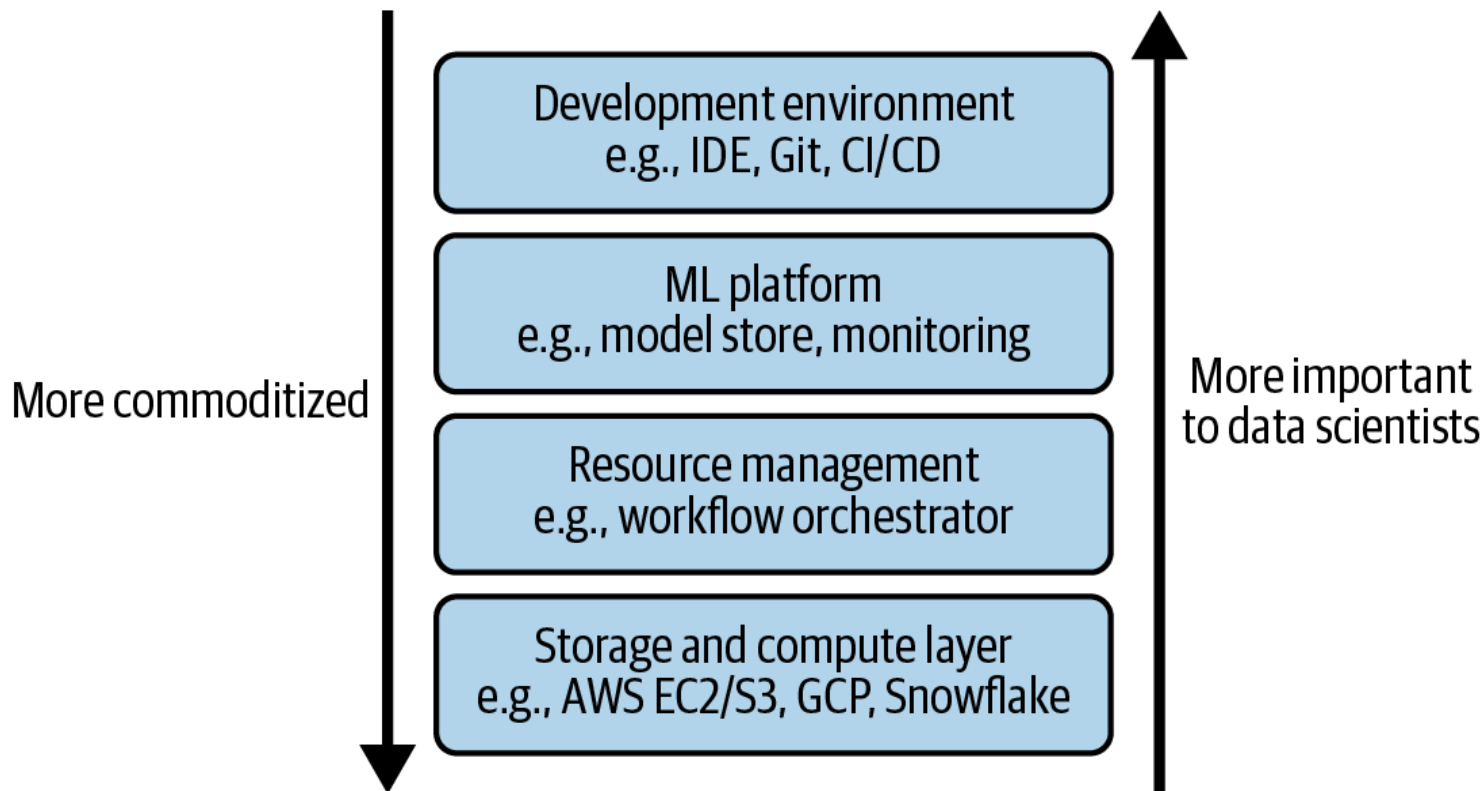
# Necesidades de infraestructura



- Depende del número y sofisticación de las apps
- El despliegue de MLOps es complejo
  - Los costes deben ser inferiores a los beneficios



# Infraestructura para MLOps



# 1) Entorno de desarrollo



- IDE, versionado y CI/CD
- Estandarización del entorno
  - *Entornos virtuales (conda, virtualenv, etc.)*
  - *requirements.txt*
  - *AWS Cloud, Amazon SageMakerStudio, GitHub Spaces...*
- Desde desarrollo (dev) hasta producción (prod)
  - ¿Como recrear las condiciones de ambos entornos?
  - Contenedores (*docker*) para ejecutar en cualquier hardware

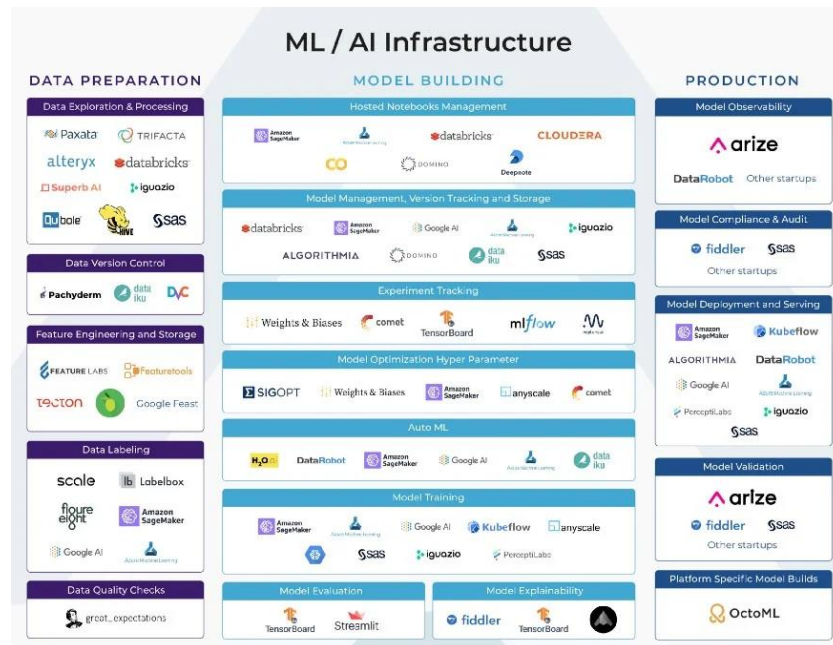
## 2) Plataformas ML



*Existen muchísimas soluciones...*

### ► **Weight & Biases**

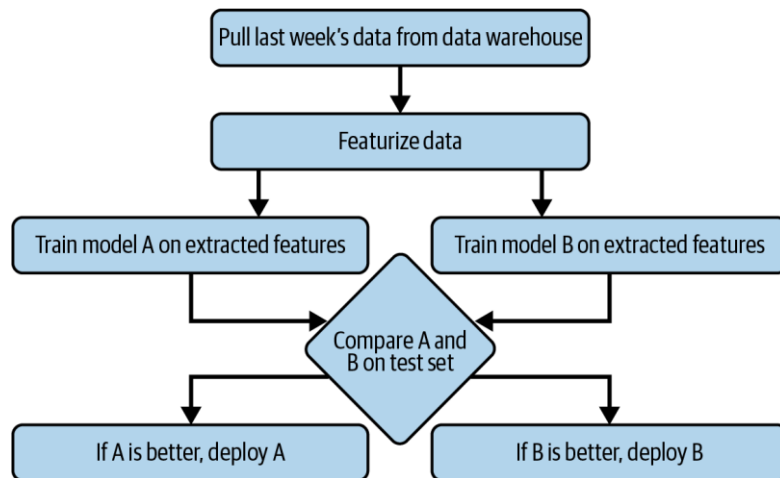
- Despliegue de modelos
- Almacenado de modelos
- Almacenado de features
- Monitorización
- Etc.



### 3) Manejo de recursos



- Manejo de las etapas de ciencia de datos
- Repetitividad y dependencias
  - ▶ *Cron, Schedulers and Orchestrators*





# 4) Almacenamiento y computación



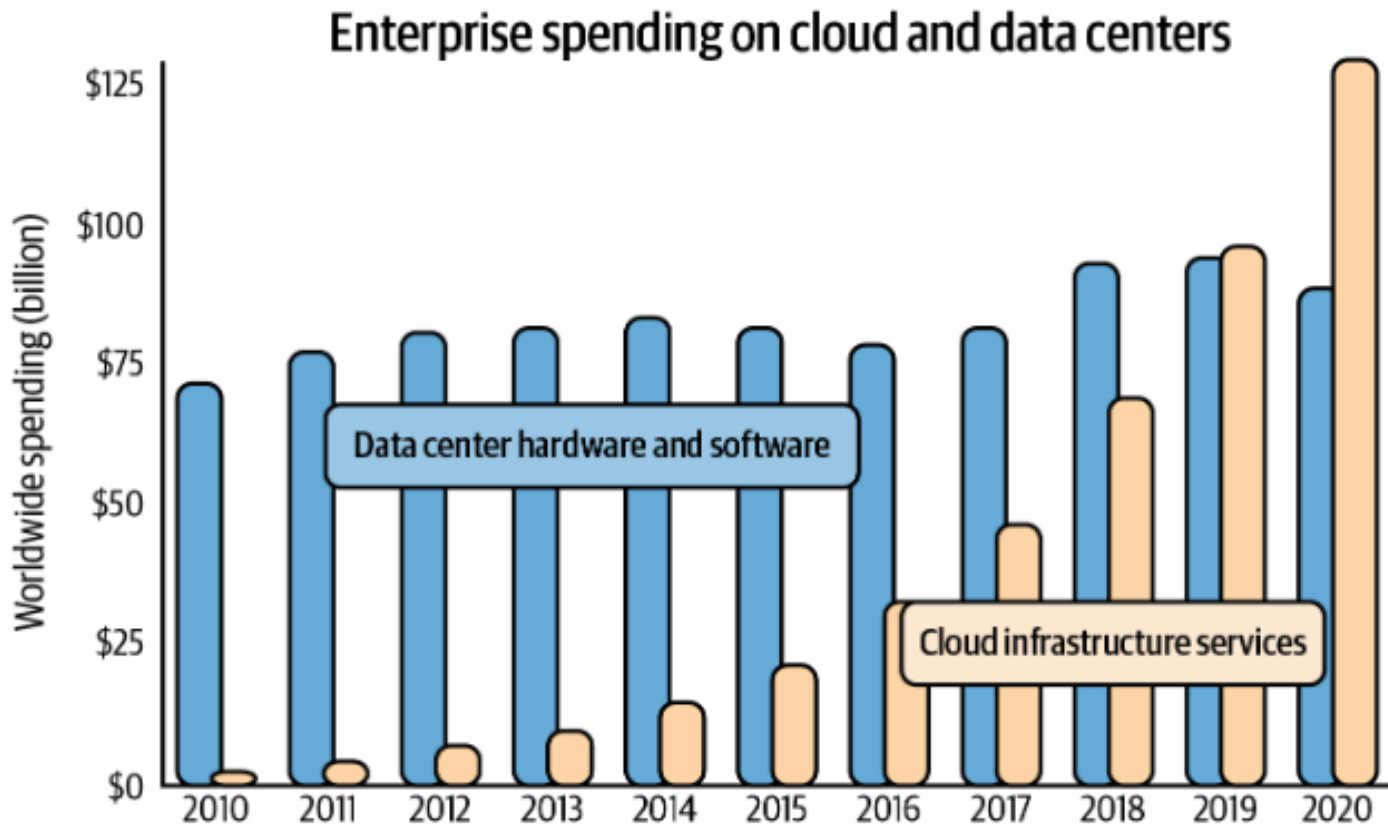
## ▸ **Almacenamiento**

- ▶ Local: disco duro o PCs
- ▶ Remoto: On-premise en datacenter privado
- ▶ Servicios cloud: Amazon S3, Snowflake...

## ▸ **Computación**

- ▶ Local: CPU
- ▶ Remoto: Servidor privado con GPUs
- ▶ Servicios cloud: AWS EC2, GCP...

## 4) Almacenamiento y computación



# Referencias

- Huyen, C. (2022). *Designing machine learning systems*. O'Reilly Media, Inc.
- Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., ... & Heidmann, L. (2020). *Introducing MLOps*. O'Reilly Media.
- Google (2024). *MLOps: Continuous delivery and automation pipelines in machine learning*.
- Stanford, CS 329S (2022). *Machine Learning Systems Design*.