

# Boston Housing

**Disponibilidad servicio para  
predecir el precio de una  
vivienda**

API • DVC • CI/CD • SHAP • Drift

Juan David Rincón

# Agenda

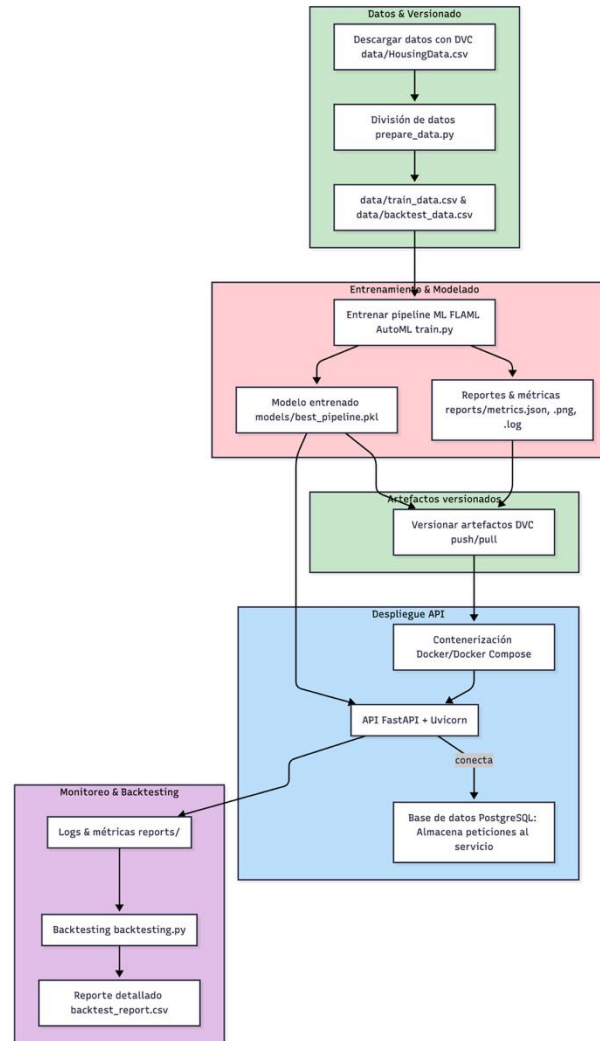
- Objetivo
- Arquitectura
- Pipeline (DVC) y entrenamiento
- API FastAPI y despliegue (Docker) --> Disponibilizar un servicio
- Evaluación y explicabilidad (SHAP)
- Monitoreo y drift
- CI/CD, retraining
- Siguiendo pasos

# Objetivo

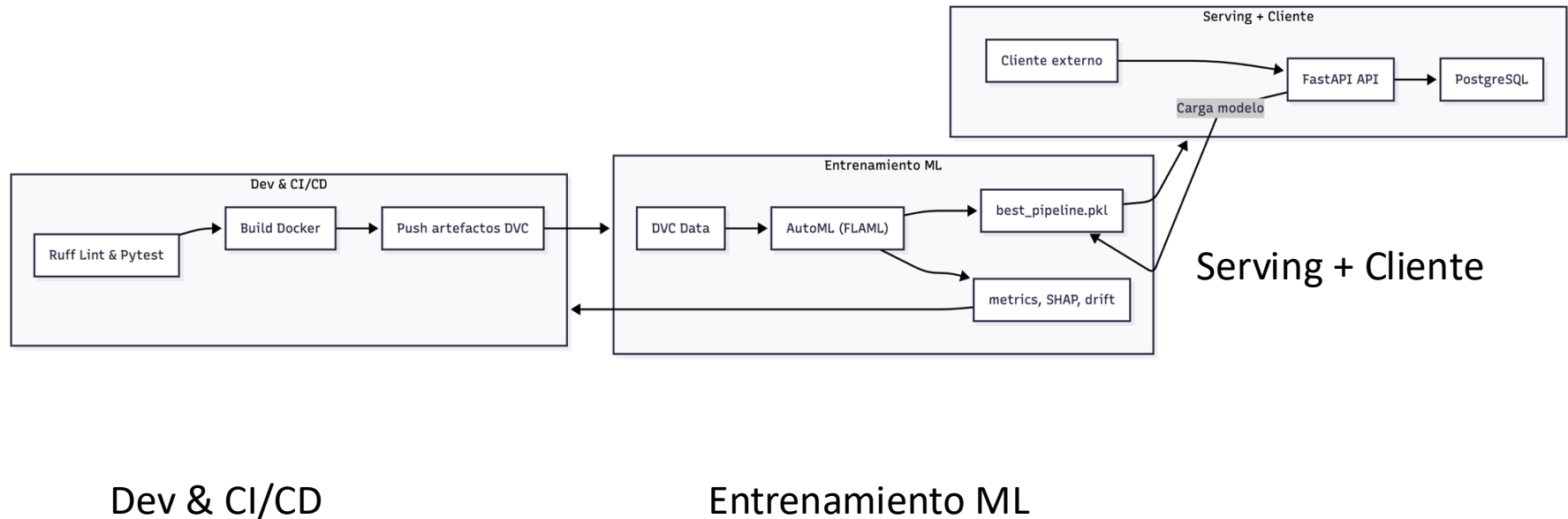
- **¿Por qué predecir el precio de una casa?**
- **Objetivo de negocio:** Un producto detrás de una idea de negocio
- **Publico Target (Magnitud):** Cantidad de solicitudes que se le realizaran al servicio.
- **Stack Tecnológico:** Herramientas que me permitan construir y disponibilidad el modelo
- **Decisiones:** ajustar tasas/descuentos, rankear oportunidades, de cara al negocio.
- **Métrica de éxito:**  $RMSE \downarrow$ ,  $R^2 \uparrow$ ,  $time-to-quote < X s$
- **Monitoreo del performance del servicio:** Tiempos de respuesta, cantidad de usuarios, latencia, control de errores,...

# Arquitectura del Proyecto

- Entreno
- Servicio
- Monitoreo
- Trazabilidad

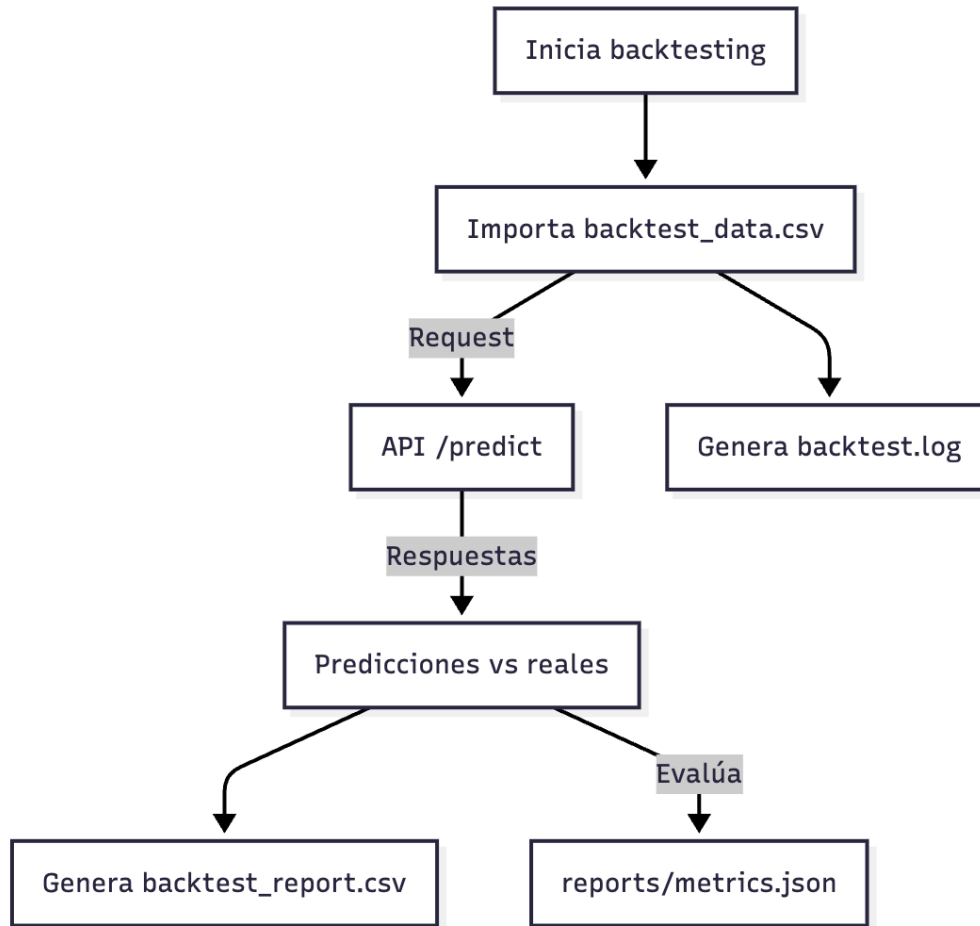


# Busqueda del Cumplimiento

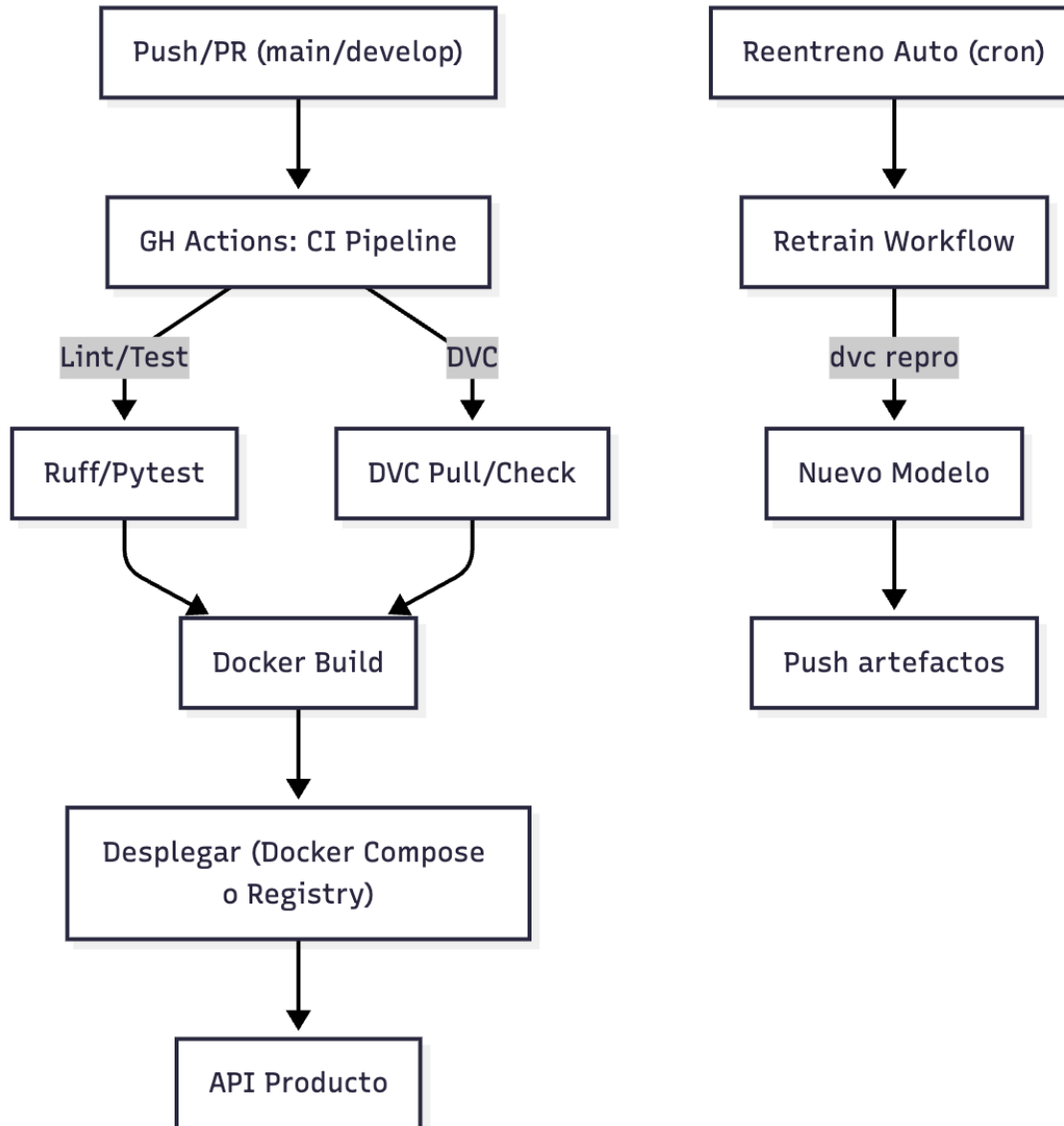


**Separación de responsabilidades:** Dev/CI para calidad y empaquetado; Entrenamiento para producir modelos **versionados**; Serving para **exponer predicciones** con trazabilidad

# Monitoreo-BackTesting



# Git Actions



# Siguientes Pasos

- Mayor monitoreo de la infraestructura y del performance del modelo.
- Nutrir la predicción con explicación de sus features por medio de los SHAP values.
- Permitir el Consumo del servicio de forma más amigable, construir un MCP que permita conectarlo a cualquier LLM y hacer predicciones con lenguaje natural.