

# Documentación Técnica del Repositorio: TFM-VALORALIA

Este documento detalla la estructura, contenido y funcionalidad de cada archivo alojado en el repositorio oficial del proyecto.

## 1. Esquema de Arquitectura del Repositorio

A continuación se presenta la estructura de directorios diseñada para garantizar la reproducibilidad y el despliegue en producción del sistema Valoralia Enterprise.

Plaintext

```
TFM-VALORALIA/
    ├── README.md          <-- Documentación general y guía de uso
    ├── pitch.html         <-- Landing Page Comercial (Capa de Negocio)
    ├── Dashboard.html     <-- Panel de Control Técnico MLOps (Capa de Ingeniería)

    ├── Notebooks/          <-- Laboratorio de Ciencia de Datos (I+D)
    │   ├── 1_Ingestion_Datos.ipynb
    │   ├── 2_Preprocesamiento_y_Calidad.ipynb
    │   ├── 3_Etrenamiento_Modelo_Hibrido.ipynb
    │   ├── 4_Interfaz_de_Tasación_Inteligente.ipynb
    │   ├── Notebook_5 - XAI_Explicabilidad.ipynb
    │   ├── 6_Test_de_Estrés_y_Concept_Drift.ipynb
    │   └── V2_VALIDACION_MERCADO.ipynb

    ├── DATA/                <-- Trazabilidad del Dato
    │   ├── datos_inmobiliarios_madrid.csv      (Raw Data)
    │   ├── dataset_valoralia_final.csv        (Gold Data)
    │   └── modelo_valoralia_final.pkl       (Modelo Serializado)

    └── APP_DEPLOY/           <-- Código de Producción (SaaS Cloud)
        ├── app.py            (Frontend + Backend Logic)
        ├── generar_modelo.py    (Script de Re-entrenamiento)
        ├── Dockerfile         (Configuración de Contenedor)
        └── requirements.txt    (Dependencias)
```

---

## 2. Descripción Detallada de Archivos y Enlaces

## A. Capa de Investigación y Desarrollo ([/Notebooks](#))

Esta carpeta contiene el ciclo de vida completo del desarrollo del modelo, desde la adquisición del dato hasta la validación avanzada.

- **Notebook 1: Ingestión de Datos**
  - *Descripción:* Script de *Web Scraping* y adquisición de datos brutos. Establece la conexión con fuentes inmobiliarias y genera el dataset inicial.
  -  [Ver Código en GitHub](#)
- **Notebook 2: Preprocesamiento y Calidad**
  - *Descripción:* Pipeline ETL (Extract, Transform, Load). Realiza limpieza de *outliers*, imputación de nulos y enriquecimiento con datos socioeconómicos del INE.
  -  [Ver Código en GitHub](#)
- **Notebook 3: Entrenamiento del Modelo**
  - *Descripción:* Entrenamiento de la arquitectura híbrida (Random Forest). Incluye la selección de hiperparámetros y la configuración del motor predictivo.
  -  [Ver Código en GitHub](#)
- **Notebook 4: Interfaz de Prototipo**
  - *Descripción:* Primera versión de la interfaz gráfica para validación local antes del despliegue en nube.
  -  [Ver Código en GitHub](#)
- **Notebook 5: XAI - Explicabilidad (Caja de Cristal)**
  - *Descripción:* Auditoría del modelo mediante técnicas de *Explainable AI*. Justifica por qué el modelo toma sus decisiones (transparencia bancaria).
  -  [Ver Código en GitHub](#)
- **Notebook 6: Test de Estrés y Concept Drift**
  - *Descripción:* Simulación de escenarios macroeconómicos (Crisis 2026, Burbuja). Evalúa la robustez del modelo ante cambios bruscos del mercado.
  -  [Ver Código en GitHub](#)
- **Notebook V2: Validación de Mercado**
  - *Descripción:* Validación adicional de métricas y alineación con precios de mercado reales.
  -  [Ver Código en GitHub](#)

## B. Capa de Negocio y Monitorización (Raíz)

Archivos orientados a la presentación comercial y control técnico del producto.

- **Pitch Comercial (Landing Page)**
  - *Descripción:* Web de presentación B2B para inversores y clientes. Actúa como punto de entrada ("Embudo de conversión") hacia la demostración técnica.
  -  [Ver Archivo](#)
- **Dashboard Técnico MLOps**
  - *Descripción:* Panel de control para el equipo de ingeniería. Monitoriza KPIs en tiempo real, latencia y alertas de *Concept Drift*.

-  [Ver Archivo](#)

## C. Capa de Despliegue en Producción ([/APP\\_DEPLOY](#))

Código fuente que se ejecuta actualmente en el servidor AWS EC2.

- **app.py (El Producto)**
  - *Descripción:* Aplicación SaaS completa desarrollada en Streamlit. Integra el modelo predictivo con la **Capa de Lógica de Negocio (Anti-Alucinaciones)** que impide inputs incoherentes por parte del usuario.
  -  [Ver Código en GitHub](#)
- **generar\_modelo.py (El Cerebro)**
  - *Descripción:* Script de entrenamiento automatizado. Contiene la lógica de re-entrenamiento con los precios suelo de mercado 2026 validados.
  -  [Ver Código en GitHub](#)
- **Dockerfile (Ingeniería Cloud)**
  - *Descripción:* Archivo de configuración para la "dockerización" de la aplicación. Permite que el sistema sea replicable y auto-escalable en cualquier servidor Linux.
  -  [Ver Código en GitHub](#)

## D. Datos y Artefactos ([/DATA](#))

Repositorio de activos de datos que garantiza la trazabilidad científica del proyecto.

- **datos\_inmobiliarios\_madrid.csv**
  - *Descripción Técnica:* **Fuente Original.** Dataset bruto obtenido mediante técnicas de *Web Scraping* con 10.000 registros iniciales sin procesar. Prueba la autoría de la ingeniería de datos.
  -  [Ver Archivo](#)
- **dataset\_valoralia\_final.csv**
  - *Descripción Técnica:* **Dataset de Entrenamiento (Gold Data).** Conjunto de datos depurado, normalizado y enriquecido con variables socioeconómicas (INE), utilizado para entrenar el modelo final.
  -  [Ver Archivo](#)
- **modelo\_valoralia\_final.pkl**
  - *Descripción Técnica:* **Modelo Serializado.** Objeto binario que contiene la arquitectura del modelo, los hiperparámetros optimizados y los pesos aprendidos, listo para inferencia en producción.
  -  [Ver Archivo](#)