

# MANUAL DE EXPLOTACIÓN



# ONDAAKIN

## Manual de Explotación – Proyecto OndAAkin

Introducción	2
Descripción general del sistema	2
Arquitectura del sistema	2
Requisitos del sistema	3
Hardware:	3
Software:	3
Despliegue de la plataforma	3
Guía de uso	4
Mantenimiento y actualizaciones	4
Resolución de problemas	5
Logs y monitoreo	5

## Introducción

Este documento describe el procedimiento de despliegue, uso, mantenimiento y resolución de incidencias de la plataforma tecnológica **0ndAAkin**, cuyo objetivo es mitigar el desperdicio alimentario mediante inteligencia artificial.

---

## Descripción general del sistema

**0ndAAkin** es un proyecto con un sistema basado en visión por computador, inteligencia artificial y analítica de datos, capaz de:

- Detectar y segmentar restos de alimentos en imágenes de platos.
  - Clasificar los tipos de comida desperdiciada.
  - Generar estadísticas automatizadas mediante dashboards.
  - Interactuar con documentación técnica mediante un sistema RAG (Retrieval Augmented Generation).
- 

## Arquitectura del sistema

### Componentes principales:

- **Frontend:** Interfaz web construida con Bootstrap.
- **Backend:** API REST con Django.
- **Modelo IA:** YOLO11-seg entrenado para segmentar y clasificar alimentos.
- **Sistema de Datos:** MongoDB y HDFS.
- **ETL y Procesamiento:** PySpark.
- **RAG:** LangChain + Chroma + deepseek-v2 (via Ollama).

- **Dashboarding:** Metabase.
  - **Infraestructura:** Docker Compose + CI/CD vía GitHub Actions.
- 

## Requisitos del sistema

### Hardware:

- CPU de al menos 4 núcleos
- 8 GB RAM mínimo
- Recomendado: GPU NVIDIA con soporte CUDA para entrenamiento y detección
- 20Gb de espacio libre

### Software:

- Docker
  - Docker Compose
  - Sistema operativo: Linux, macOS o Windows (WSL2 para este último)
  - Git
- 

## Despliegue de la plataforma

1. Clonar el repositorio:

```
git clone https://github.com/amaiaDi/7_Reto2_0ndAAkin_G3.git  
cd 7_Reto2_0ndAAkin_G3/Despliegue
```

2. Ejecutar script dentro de la carpeta despliegue del proyecto:

```
sudo sh start.sh
```

3. Acceder a los servicios en los siguientes puertos:

- Página principal: <http://localhost:43000>
  - Web App: <http://localhost:43001> (admin | Password0)
  - Metabase: <http://localhost:43003> (admin@admin.com | 12345@a)
  - HDFS UI: <http://localhost:43007>
- 

## Guía de uso

- **Carga de imágenes:** A través del frontend.
  - **Visualización de resultados:** Dashboards en Metabase.
  - **Consulta vía RAG:** Interfaz conversacional integrada.
  - **Logs:** Accesibles en el contenedor correspondiente vía `docker logs`.
- 

## Mantenimiento y actualizaciones

- **Actualización del modelo IA:** Entrenamiento offline y actualización de contenedor.
- **CI/CD:** Automático vía GitHub Actions al hacer push en rama `main`.
- **Backup de datos:** Usar scripts de exportación desde MongoDB/HDFS.

## Resolución de problemas

Problema	Causa probable	Solución
Contenedor no levanta	Puerto ocupado o error en .env	Revisar logs con <code>docker logs [contenedor]</code>
Metabase no carga	Errores en HDFS o red	Verificar conexión entre contenedores
RAG no responde	Servicio Ollama no activo	Asegurar que el modelo deepseek esté corriendo o el pull del modelo correctamente ejecutado

---

## Logs y monitoreo

- Logs accesibles por `docker logs`
- Se recomienda integrar con Prometheus + Grafana para monitoreo extendido