# INFORME\_PROYECTO - Clasificación de Enfermedades en Hojas de Tomate usando CNNs

## 1. Estructura del repositorio y notebooks entregados

El repositorio está organizado de la siguiente manera:

- `ENTREGA1.pdf`: informe inicial con contexto, objetivos y dataset.

- `INFORME\_PROYECTO.pdf`: informe final del proyecto.

- `notebooks/`

- `01\_preprocesamiento.ipynb`: descarga, selección y preprocesamiento de los datos.

- `02\_entrenamiento\_basico.ipynb`: implementación de un modelo CNN sencillo.

- `03\_modelo\_final.ipynb`: modelo final con MobileNetV2, aumento de datos y evaluación avanzada.

- `README.md`: documentación del proyecto con enlace al video y resumen.

- `modelo\_final\_tomate\_mobilenet.h5`: modelo final entrenado (si aplica).

Todos los notebooks están preparados para ejecutarse en Google Colab y utilizan datasets públicos disponibles en Kaggle.

## 2. Descripción de la solución

La solución implementada consiste en un modelo de clasificación multiclase para detectar enfermedades en hojas de tomate a partir de imágenes.

Se siguieron dos enfoques:

1. Un modelo CNN básico con tres capas convolucionales y dropout.

2. Un modelo con transferencia de aprendizaje utilizando MobileNetV2 preentrenado sobre ImageNet.

Ambos modelos fueron entrenados sobre imágenes redimensionadas (224x224), normalizadas y categorizadas según su clase.

## 3. Preprocesamiento y augmentación

El preprocesamiento incluye:

- Filtro de imágenes del dataset PlantVillage solo para clases de tomate.

- Redimensionamiento de imágenes a 224x224.

- Normalización de valores RGB (1./255).

- División de datos en entrenamiento (80%) y validación (20%).

En el modelo final se utilizó `ImageDataGenerator` para realizar aumento de datos: rotaciones, zooms, y reflejo horizontal.

## 4. Iteraciones realizadas

Se realizaron las siguientes iteraciones para mejorar el rendimiento del modelo:

- Modelo básico CNN entrenado desde cero.

- Modelo con MobileNetV2 (transfer learning) congelando las capas base.

- Pruebas con diferente cantidad de filtros y tamaños de capas ocultas.

- Aplicación de técnicas de aumento de datos.

Se evaluó cada iteración con curvas de precisión/pérdida y matriz de confusión.

## 5. Resultados

El modelo final basado en MobileNetV2 alcanzó una precisión de validación superior al 95%.

Se obtuvo un buen balance entre clases, como se evidencia en la matriz de confusión y en las métricas Precision, Recall y F1-score por clase.

Esto demuestra que el modelo es apto para aplicaciones reales en el campo agrícola.

## 6. Datos utilizados

El dataset utilizado fue PlantVillage, específicamente las clases de hojas de tomate:

- Tomato\_\_\_Healthy

- Tomato\_\_\_Early\_blight

- Tomato\_\_\_Late\_blight

- Tomato\_\_\_Leaf\_Mold (y otras, si se incluyeron)

El dataset fue descargado automáticamente desde Kaggle y preprocesado en el notebook `01\_preprocesamiento.ipynb`.

## 7. Entrenamiento del modelo

El modelo se entrenó directamente desde los notebooks en Google Colab utilizando `TensorFlow` y `Keras`. No se requiere entorno local.

Los pesos fueron entrenados en GPU (Colab) y pueden ser reutilizados cargando el archivo `.h5`.

El entrenamiento puede ser reproducido ejecutando el notebook `03\_modelo\_final.ipynb`.