

# Documentación Técnica: Uso de GridFS en el Proyecto de IA

---

## Objetivo

Esta API permite almacenar imágenes y temperaturas generadas por un sistema de inteligencia artificial, utilizando MongoDB como motor de base de datos. Se aprovecha GridFS para gestionar imágenes de gran tamaño y se almacena la información de temperatura en una colección separada.

## Tecnologías Utilizadas

- Python 3.x
- Flask – Framework web ligero para crear APIs REST
- MongoDB – Base de datos NoSQL para almacenar imágenes y datos asociados
- GridFS (pymongo.gridfs) – Sistema de archivos para almacenar archivos binarios en MongoDB
- BSON / ObjectId – Identificadores únicos de MongoDB
- IO / BytesIO – Lectura y envío de archivos binarios en memoria

## Arquitectura de la Base de Datos

### ◆ Base de Datos: imgsdb

Contiene las imágenes almacenadas en formato binario usando GridFS.

### ◆ Base de Datos: tempsdb

Contiene la colección temperatures, donde se almacenan las temperaturas capturadas en cada registro.

## Endpoints Disponibles

### 1. POST /upload

Sube una imagen y su temperatura asociada.

Parámetros (multipart/form-data):

- image: archivo de imagen (.jpg, .png, etc.)
- temperature: valor numérico de temperatura (ej. 35.4)

- Alternativamente, también puede recibir imágenes codificadas en Base64, que deben ser decodificadas antes de su almacenamiento.

Proceso:

- La imagen se guarda en imgsdb con GridFS.
- La temperatura se almacena en la colección temperatures de tempsdb.

Respuesta Exitosa (200):

```
{
  "message": "Imagen y temperatura almacenadas exitosamente",
  "file_id": "665a63f2c14e7f01b7a32abc"
}
```

## 2. GET /image/<file\_id>

Descarga una imagen por su ID.

Parámetro de ruta: file\_id (ej. 665a63f2c14e7f01b7a32abc)

Respuesta Exitosa: Imagen en binario (MIME image/jpeg)

Respuesta de Error (404):

```
{
  "error": "No se encontró la imagen"
}
```

## 3. GET /images

Lista todas las imágenes almacenadas con su información relevante.

Respuesta Exitosa (200):

```
[
  {
    "file_id": "665a63f2c14e7f01b7a32abc",
    "filename": "img01.jpg",
    "temperature": null,
    "timestamp": "2024-06-01T23:45:30.000Z"
  },
  ...
]
```