ANÁLISIS DE LA PROGRAMACIÓN A REALIZAR

– Juan Carlos Alonso Hernando –

30 / 01 / 2026

1. ESTRUCTURA GENERAL DEL PROYECTO

La idea principal es desarrollar una aplicación web que actúe como un integrador de datos. Voy a recolectar información de tres fuentes distintas (CSV, JSON y XML) para consolidarlas en un único origen de datos no relacional utilizando **MongoDB**.

El proyecto se dividirá en los siguientes componentes lógicos:

- **Vistas (HTML/PHP):**
  - index.php: El archivo principal que mostrará el listado completo de alumnos.
  - formulario.php: formulario que servirá tanto para dar de alta nuevos alumnos como para editar los existentes.
- **Lógica de Control (PHP):**
  - importar.php: Script encargado de la lectura de ficheros externos y el volcado inicial a la base de datos.
  - guardar.php y borrar.php: Controladores para gestionar las operaciones CRUD individuales.
  - borrar_todo.php: Un botón para eliminar la colección completa.
- **Capa de Datos:**
  - db.php: Archivo de configuración de la conexión al Driver nativo de MongoDB.
- **Diseño (CSS):**
  - style.css: Archivo para diseñar los estilos de la aplicación.

2. CAPA DE DATOS

La estrategia de datos se centra en la flexibilidad de MongoDB. A diferencia de SQL, voy a utilizar una estructura de **Documentos Embebidos**.

Estrategia de Almacenamiento:

1. **Fuentes de entrada:** Voy a procesar tres ficheros en una carpeta /data: datos.csv, datos.json y datos.xml.
2. **Estructura del Documento:** Cada entrada en la colección filas tendrá esta estructura:
    - Numero: (Int) Identificador de lista.
    - Alumno: (Objeto) Contendrá Nombre, Apellidos, Sexo y es_profe_sexi (Boolean).

3. CAPA DE NEGOCIO (LÓGICA)

En esta capa voy a gestionar la transformación de los datos y el ciclo de vida de los registros:

- **Normalización:** Al leer los archivos, voy a forzar los tipos de datos (casting). Por ejemplo, los valores de "profe sexy" que vengan como texto "0" o "false" se convertirán a booleanos reales.
- **Operaciones CRUD:**
    - **Inserción Masiva:** Voy a utilizar la clase BulkWrite para que la importación inicial sea atómica y eficiente.
    - **Upsert:** El controlador de guardado decidirá si ejecutar una actualización o una inserción basándose en si existe o no el ObjectId.

4. CAPA DE PRESENTACIÓN (INTERFAZ)

- **UX Dinámica:** Se implementarán avisos de confirmación en JavaScript antes de realizar borrados para evitar la pérdida accidental de datos.
- **Visualización:** El listado principal utilizará un cursor ordenado por el campo Numero. Se aplicarán estilos CSS para diferenciar visualmente a los alumnos marcados como "profe sexy".