

# Trabajo Práctico de Desarrollo e Investigación

**Objetivo:** El objetivo de este trabajo es investigar, diseñar y desarrollar un servidor utilizando Node.js y Express, junto con el motor de plantillas EJS, para crear un sistema de gestión de imágenes con operaciones CRUD (Crear, Leer, Actualizar y Borrar). Además, se busca integrar el servicio Cloudinary para el almacenamiento y manipulación de imágenes. El proyecto también involucra la implementación de una base de datos MySQL para el seguimiento y almacenamiento de información relacionada con las imágenes, incluida la compatibilidad con la librería express-fileupload para la carga de archivos.

**Aclaracion:** se deberán guardar las rutas de las imágenes en la base de datos.

## Entrega

1. Repositorio con el servidor creado, el cual contendrá dos EndPoint(rutas), uno que permitirá subir los archivos por medio de **express-fileupload** y otro con **cloudinary**.
2. **Fecha de entrega: 11-08-2023 - 17hs**

## Consignas

### Parte 1: Configurar un servidor desde cero:

Se deberá configurar un servidor como ya se venía trabajando en los trabajos prácticos anteriores, configurando el motor de plantillas EJS, variables de entorno, y demás paquetes necesarios.

### Parte 2: Implementar express-fileupload

1. Instalar el paquete **express-fileupload** utilizando npm.
2. Crear una ruta en tu aplicación para manejar las solicitudes de subida de archivos.
3. Implementar el manejo de subida de archivos utilizando **express-fileupload**.
4. Al subir un archivo, guardarlo en una carpeta local en el servidor.
5. Crear la vista utilizando el motor de plantilla EJS con un formulario para permitir a los usuarios seleccionar y subir archivos al servidor.
6. Se deberán manejar los posibles errores y proporcionar retroalimentación al usuario después de la subida.

### Parte 3: Implementar cloudinary

1. Crear una cuenta gratuita en [Cloudinary](#) si no lo tienen.
2. Instalar el paquete **cloudinary** utilizando npm.
3. Configurar la aplicación Node.js para utilizar las credenciales proporcionadas por Cloudinary.
4. Crear una ruta en la aplicación para manejar las solicitudes de subida de archivos.
5. Implementar el manejo de subida de archivos a Cloudinary utilizando el paquete **cloudinary**.
6. Crear la vista utilizando el motor de plantilla EJS con un formulario para permitir a los usuarios seleccionar y subir archivos al servidor.
7. Después de la subida exitosa, mostrar los enlaces a las imágenes subidas desde Cloudinary.