

**Universidad San Carlos de Guatemala**

**Facultad de Ingeniería**

**Escuela de Ciencias y Sistemas**

**Sistemas de Bases de Datos 2**



## **PROYECTO 2**



# Proyecto: Sistema de Gestión de Biblioteca con MongoDB

## Descripción del Proyecto

Los estudiantes desarrollarán un sistema de gestión de biblioteca utilizando MongoDB como base de datos. Este sistema incluirá un backend en Node.js con Express para manejar las operaciones CRUD a través de una API RESTful, y un frontend utilizando HTML, CSS y JavaScript para interactuar con la API y proporcionar una interfaz de usuario amigable.

## Objetivos del Proyecto

- Comprender y aplicar los conceptos de MongoDB en un entorno práctico.
- Desarrollar un backend que implemente las operaciones CRUD usando una API RESTful.
- Crear un frontend que consuma la API y permita a los usuarios interactuar con el sistema de gestión de la biblioteca.
- Familiarizarse con la estructura de documentos y colecciones en MongoDB.
- Aprender a utilizar operaciones de agregación para obtener información útil.

## Requerimientos

### 1. Entorno de Desarrollo:

- MongoDB instalado (local o en la nube).
- Node.js y npm instalados para el desarrollo del backend.
- Herramientas para interactuar con MongoDB (MongoDB Compass o línea de comandos).
- Conocimientos básicos de HTML, CSS y JavaScript para el desarrollo del frontend.

### 2. Modelo de Datos:

- Crear las siguientes colecciones:

#### ■ Libros

- `titulo`: String
- `autor_id`: ObjectId (referencia a la colección de autores)
- `anio_publicacion`: Number
- `genero`: String
- `disponibilidad`: Boolean

#### ■ Autores

- `nombre`: String
- `apellido`: String
- `nacionalidad`: String

#### ■ Usuarios

- `nombre`: String
- `apellido`: String
- `email`: String
- `fecha_registro`: Date

## Actividades

### 1. Backend - API RESTful:

- **Configurar el servidor:**
  - Configurar un servidor utilizando Node.js y Express.
  - Conectar el servidor a la base de datos MongoDB usando Mongoose.
- **Implementar las rutas:**
  - Crear rutas para cada operación CRUD en las colecciones (libros, autores, usuarios).
  - Implementar validaciones para los datos entrantes en las rutas de creación y actualización.
- **Operaciones CRUD:**
  - **Crear:**
    - Implementar rutas para agregar nuevos libros, autores y usuarios.
  - **Leer:**
    - Implementar rutas para obtener todos los libros, autores y usuarios, así como consultas específicas (por autor, disponibilidad, etc.).
  - **Actualizar:**
    - Implementar rutas para modificar información de libros, autores y usuarios.
  - **Eliminar:**
    - Implementar rutas para eliminar libros, autores y usuarios.
- **Operaciones de Agregación:**
  - Crear una ruta que devuelva la cantidad de libros por autor.
  - Generar una ruta que muestre una lista de autores y el número total de libros escritos por cada uno.

### 2. Frontend:

- **Estructura HTML:**
  - Crear páginas HTML para la visualización de libros, autores y usuarios.
  - Incluir formularios para agregar y editar libros, autores y usuarios.
- **Estilo CSS:**
  - Estilizar las páginas utilizando CSS para hacerlas visualmente atractivas y fáciles de usar.
- **JavaScript:**
  - Usar JavaScript para consumir la API RESTful.
  - Implementar funcionalidades para mostrar la lista de libros, autores y usuarios.
  - Implementar funciones para manejar la creación, actualización y eliminación de registros utilizando las rutas del backend.
  - Manejar la respuesta de la API y actualizar la interfaz de usuario en consecuencia.

### 3. Consultas Avanzadas:

- **Buscar libros por género:**
  - Consultar todos los libros que pertenezcan a un género específico (e.g., "Ficción", "Historia").
- **Filtrar libros por rango de años:**
  - Obtener todos los libros publicados entre dos años específicos (e.g., entre 1990 y 2005).
- **Buscar libros por título parcial:**
  - Implementar una búsqueda que devuelva todos los libros cuyos títulos contengan una palabra clave o fragmento.
- **Listar autores de una nacionalidad específica:**
  - Consultar todos los autores que sean de una determinada nacionalidad (e.g., "Español").
- **Mostrar usuarios registrados en un rango de fechas:**
  - Obtener todos los usuarios que se hayan registrado entre dos fechas específicas.
- **Buscar autores que han escrito más de 'N' libros:**
  - Devolver una lista de autores que han escrito más de una cantidad determinada de libros (e.g., más de 5 libros).
- **Listar los 5 libros más recientes:**
  - Mostrar los cinco libros más recientes, basados en el año de publicación.
- **Obtener libros no disponibles:**
  - Consultar todos los libros que actualmente no están disponibles para préstamo.
- **Contar el número total de usuarios:**
  - Devolver el número total de usuarios registrados en la biblioteca.
- **Listar autores con más de una nacionalidad:**
- Mostrar una lista de autores que tengan más de una nacionalidad (esto requerirá modificar el modelo de datos para permitir múltiples nacionalidades).
- 
- 4. **Actualizaciones y Eliminaciones:**
  - Actualizar la disponibilidad de un libro (cambiar de disponible a no disponible y viceversa).
  - Modificar la nacionalidad de un autor.
  - Cambiar el email de un usuario.
  - Eliminar un libro de la colección.
  - Eliminar un autor (asegurarse de manejar correctamente las referencias en la colección de libros).
  - Eliminar un usuario que no haya realizado ninguna transacción en la biblioteca.

## ENTREGA

- ❖ La entrega será el 25 de Octubre, a más tardar a las 11:59 pm.
- ❖ Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio, nombre del archivo BD2\_Proyecto2\_#Carnet|Grupo.
- ❖ Copias totales o parciales entre integrantes del laboratorio o bajadas de internet obtendrán nota de 0 puntos.
- ❖ No se recibirán entregas tardes.
- ❖ Agregar al respectivo auxiliar al repo.