

Práctica de MongoDB y Mongoose

Objetivo:

Se deberán implementar colecciones con propiedades embebidas y referenciadas. Además, será necesario realizar un CRUD completo.

Criterios de Evaluación

1. Presentación del código

- El código debe estar limpio, ordenado y bien indentado.
- Uso obligatorio de try-catch en todos los controladores para un manejo adecuado de errores.
- Organización correcta del proyecto en carpetas:
 - **src/config** → configuración (ej. conexión a BD).
 - **src/models** → definición de modelos Sequelize.
 - **src/routes** → definición de rutas.
 - **src/controllers** → controladores.
- Uso exclusivo de ESM modules (import / export).
- El código debe ser funcional, modularizado y sin errores de ejecución.

2. Crud y colecciones (con MongoDB + Mongoose)

- Uso correcto de Mongoose con conexión a MongoDB, definición adecuada de esquemas y relaciones.
- Respuestas claras y códigos HTTP apropiados en cada operación:
 - **201 Created** → al crear.
 - **200 OK** → en consultas y actualizaciones exitosas.
 - **500 Internal Server Error** → errores inesperados.
- Uso de populate en consultas para traer datos relacionados.
- Manejo de eliminaciones lógicas y eliminaciones en cascada. Mostrar mensajes de éxito o error al realizar operaciones CRUD.

Entrega mediante Git y GitHub – Uso de ramas y commits

Requisitos obligatorios para el control de versiones.

Teniendo en cuenta que en la práctica anterior las ramas **main** y **develop** contienen el mismo contenido, ahora se debe hacer lo siguiente:

- Crear una rama a partir de “**develop**” llamada “**mongoose-basico**”.
- Realizar el desarrollo del trabajo en la rama “**mongoose-basico**”.
- Al finalizar el trabajo:
 - Hacer un merge limpio de “**mongoose-basico**” hacia “**develop**”, sin conflictos.
 - Luego hacer un merge limpio de “**develop**” hacia “**main**”, para que ambas ramas queden sincronizadas nuevamente
- Durante el desarrollo en la rama **mongoose-basico**, se deben realizar al menos 5 commits con mensajes claros y descriptivos.

Requisitos obligatorios del repositorio remoto

- El nombre del repositorio en GitHub debe seguir la siguiente convención:
 - mongoose-apellido-nombre. Ejemplo: mongoose-perez-juan
- El repositorio debe contener:
 - Todos los archivos del proyecto (app.js, package.json, src/ completo).
 - El archivo .env.example con los nombres de las variables utilizadas.
- El archivo .env y la carpeta node_modules no deben subirse.
- El enlace al repositorio debe entregarse en Google Classroom.

Consignas:

1. Crear colecciones

- Deben crear mínimo 3 colecciones a elección.
- Cada proyecto debe cumplir:
 - 1 propiedad embebida en alguna colección.
 - 1 relación 1:1.
 - 1 relación 1:N.
 - 1 relación N:M.
- Justificar en un documento o en el README.md por qué eligieron embebido o referenciado en cada caso.

2. Crud Básico

Implementar creación, lectura, actualización y eliminación en todas las colecciones creadas.

- Uso de try/catch en todos los controladores.
- La eliminación debe contemplar lógica y en cascada.

3. Investigar

Se debe investigar, aplicar y justificar:

- Como se utiliza el **populate** desde las colecciones que no tienen referencias.
- Como se realizan las eliminaciones lógicas y en cascada.
- Cómo crear un endpoint que permita agregar un nuevo vínculo en una relación muchos a muchos.

Bonus: Implementación de validaciones básicas o utilizando express-validator.