

La separación entre **modelos** y **servicios** no es redundante, sino una práctica que promueve la modularidad y el principio de responsabilidad única. Los modelos se encargan exclusivamente de interactuar con los datos, mientras que los servicios manejan la lógica de negocio y orquestan operaciones más complejas. Esta división permite que el código sea más fácil de mantener, probar y escalar, especialmente en aplicaciones que crecen en complejidad.

---

## Ejercicio Práctico

### Organización de la Capa de Modelos en tu API

Sabrina y Matías regresan para revisar tu progreso. Esta vez, Sabrina es quien toma la palabra con entusiasmo:

"¡Tu organización ha mejorado muchísimo! Ya tienes rutas bien definidas, controladores claros y una capa de servicios eficiente. Ahora es momento de llevar la estructura un paso más allá."



Matías asiente y añade con una mirada seria:



"Hasta ahora, has estado manejando datos simulados directamente en la capa de servicios, pero en un proyecto real, los datos suelen almacenarse en bases de datos o archivos estructurados. Por eso, vamos a introducir la capa de modelos, que centralizará la gestión de datos."

### Misión:

#### 1. Crear la capa de modelos

- Organiza los datos de tu aplicación en archivos JSON dentro de una carpeta llamada **data**.
- Migra los datos simulados que estaban en los servicios hacia estos archivos.
- Asegúrate de que la estructura de los archivos JSON sea clara y representativa de la información que manejas.
- Crea archivos para los modelos de tu aplicación y crea los métodos necesarios para interactuar con los JSON de datos.

## 2. Modificar los servicios para interactuar con los modelos

- Ajusta la capa de servicios para que en lugar de devolver datos directamente desde el código lo haga utilizando los métodos creados en los modelos.
- Asegúrate que los controladores sigan obteniendo los datos correctamente desde los servicios.



Matías concluye con una mirada confiada:

"Si logras completar este desafío, habrás dado un paso gigante hacia el desarrollo backend profesional. ¡Es hora de escribir código limpio y estructurado!"

---

## Materiales y Recursos Adicionales:

- Documentación Oficial de JSON: <https://www.json.org/json-es.html>
- Documentación de Node.js sobre el módulo FS: <https://nodejs.org/api/fs.html>

---

## Preguntas para Reflexionar:

- ¿Cuál es la importancia de la capa de modelos en una API REST y cómo contribuye a la organización del código?
- ¿Qué ventajas tiene utilizar archivos JSON como base de datos en lugar de manejar datos directamente en los servicios?

- ¿Qué posibles problemas podrías encontrar al usar archivos JSON como almacenamiento de datos y cómo podrías solucionarlos?
  - ¿Cómo podrías escalar la solución actual si en el futuro decides migrar a una base de datos relacional o NoSQL?
- 

## Próximos Pasos:

- **Datos en la nube:** Configurando y accediendo a datos en un servidor externo.
- **Autenticación y Autorización:** Manejando el acceso público y privado de nuestros datos.
- **Fin de la cursada:** daremos cierre al curso mediante la presentación de proyectos finales.