# **📝 Trabajo Práctico: API REST para To-Do List con Node.js, Express y Mongoose**

## **🎯 Objetivo**

Construir una API RESTful utilizando **Node.js**, **Express** y **Mongoose** que permita administrar tareas (Task), organizarlas en Sprints, y tener un Backlog general. La información se almacenará en una base de datos MongoDB.

## **📦 Requisitos del Proyecto**

### **1. Estructura del Proyecto**

Organizar el proyecto en carpetas siguiendo buenas prácticas. Por ejemplo:

/todo-api

├── models/

│ ├── Task.js

│ ├── Sprint.js

│ └── Backlog.js

├── routes/

│ ├── tasks.js

│ ├── sprints.js

│ └── backlog.js

├── controllers/

│ ├── taskController.js

│ ├── sprintController.js

│ └── backlogController.js

├── app.js

└── .env

## **🧱 Modelos (Guía de implementación)**

### **Task**

Modelar una tarea con los siguientes campos:

* Un título obligatorio de tipo texto.
* Una descripción opcional.
* Un estado que solo puede tomar ciertos valores (por ejemplo, pendiente, en progreso o completado).
* Una fecha límite obligatoria.
* Un color para identificar visualmente la tarea.

### **Sprint**

Un sprint agrupa un conjunto de tareas durante un período de tiempo definido. Debe tener:

* Una fecha de inicio obligatoria.
* Una fecha de cierre obligatoria.
* Una lista de referencias a tareas.
* Un color para diferenciar visualmente cada sprint.

### **Backlog**

El backlog representa una lista general de tareas no asignadas aún a ningún sprint. Debe incluir:

* Una lista de tareas que todavía no están asignadas a un sprint. Estas deben estar relacionadas con el modelo de tarea.

## **🔧 Endpoints Requeridos**

### **🔹 Tasks**

* GET /tasks: Obtener todas las tareas
* GET /tasks/:id: Obtener una tarea por ID
* POST /tasks: Crear una tarea
* PUT /tasks/:id: Editar una tarea
* DELETE /tasks/:id: Eliminar una tarea

### **🔹 Sprints**

* GET /sprints: Obtener todos los sprints
* GET /sprints/:id: Obtener un sprint por ID
* POST /sprints: Crear un sprint
* PUT /sprints/:id: Editar un sprint
* DELETE /sprints/:id: Eliminar un sprint
* PUT /sprints/:id/add-task/:taskId: Agregar una tarea a un sprint

### **🔹 Backlog**

* GET /backlog: Obtener el backlog
* POST /backlog: Crear backlog (solo uno en este caso)
* PUT /backlog/add-task/:taskId: Agregar una tarea al backlog

## **🛡 Validaciones**

* No se puede agregar una tarea inexistente a un sprint o backlog.
* No se puede eliminar una tarea si está asignada a un sprint.

## **🔁 Extras**

* Filtro de tareas por estado (/tasks?estado=pendiente)
* Ordenar tareas por fechaLimite
* Middleware de validación con express-validator
* Documentar la API con Swagger