

Sistema de gestión de tareas

Deberás realizar una API REST que permita el funcionamiento de un sistema de gestión de tareas, utilizando las tecnologías de Node.js mediante un framework. El proyecto tendrá que estar en un repositorio público para permitir su consulta.

El sistema de gestión de tareas tiene como propósito el que un usuario pueda visualizar, agregar, editar o eliminar las tareas que requiera registrar en su día a día. Y debido a que el sistema es utilizado por varios usuarios, es necesario que la API identifique al usuario o sesión que está solicitando cierta información. (No es necesario crear un módulo de registro y autenticación de usuarios).

Cada tarea debe tener:

- Título (Obligatorio)
- Descripción (Obligatorio)
- Estatus de compleción (Obligatorio)
- Fecha de entrega (Obligatorio)
- Comentarios (Opcional)
- Responsable (Opcional)
- Tags (Opcional)

Endpoints mínimos necesarios disponibles en la API:

- GET -> Regresa información breve de todas las tareas
- GET -> Regresa toda la información de una tarea
- POST -> Crear una tarea
- PUT -> Editar una tarea
- DELETE -> Borrar una tarea

Aspectos a evaluar:

- Funcionamiento de la API REST
- Utilización de Node.js y Express
- Estructura del proyecto
- Documentación del código y endpoints
- Buenas prácticas de programación

Se sugiere utilizar el siguiente stack de herramientas

1. Nest.js o Express.js
2. MySQL
3. TypeOrm

Retos extra (Opcionales)

1. Si deseas conseguir una puntuación extra en tu ejercicio puedes agregarle un grado de dificultad al agregar las siguientes funcionalidades.

En cada tarea agregar:

- ¿Es pública? (Obligatorio)
- Archivo (Opcional, no mayor a 5MB y sólo formatos .pdf, .png y .jpg)

Consideraciones:

1. Un usuario puede ver toda la lista de tareas públicas, así como editarlas, completarlas y eliminarlas
2. Los métodos GET deben funcionar para fines de paginación en el Frontend y mostrar cuántos resultados encontró en cada llamada en la misma respuesta
3. El sistema debe guardar una bitácora de todos los movimientos

Nota extra:

- No es obligatorio desarrollar este reto extra ni completarlo en su totalidad.

2. Como segunda opción también puedes realizar lo siguiente:

- Autenticación y Autorización: Agregar un sistema básico de autenticación utilizando tokens de acceso y así mantener rutas privadas y públicas
- Paginación, Ordenamiento y Filtrado Avanzado: filtrar tareas según diferentes criterios, como estatus de completitud, fecha de entrega, responsable, tags, etc.
- Tener por lo menos una tabla menu (de los campos opcionales) y consultar información relacionando tablas
- Montar Backend en un hosting (heroku, railway)