

## RESTful API - Sistema de Gestión de Tareas

Ejercicio práctico para posición de *Desarrollador Backend*

Deberás crear una API REST para un sistema de gestión de tareas. A través de este sistema un usuario podrá *visualizar, agregar, editar o eliminar* sus tareas personales y colectivas. El sistema será utilizado por varios usuarios, por lo que podrán definir para cada tarea si es *pública, privada o compartida*.

Cada tarea debe tener:

1. Título (Obligatorio)
2. Descripción (Obligatorio)
3. Estatus de compleción (Obligatorio)
4. Fecha de entrega (Obligatorio)
5. ¿Es pública? (Obligatorio)
6. Compartido con (Opcional, pueden ser varios usuarios)
7. Comentarios (Opcional)
8. Creado por (Obligatorio)
9. Responsable (Opcional, sólo 1 usuario)
10. Tags (Opcional)
11. Archivo (Opcional, no mayor a 5MB y sólo formatos .PDF; .PNG y .JPG)

Endpoints mínimos necesarios disponibles en la API:

1. GET -> Regresa información breve de todas las tareas
2. GET -> Regresa toda la información de una tarea
3. POST -> Crear una tarea
4. PUT -> Editar una tarea
5. DELETE -> Borrar una tarea

Consideraciones y requisitos adicionales:

1. El sistema debe guardar una bitácora de todos los movimientos realizados
2. Un usuario puede crear tareas y asignar como responsable a otro usuario; siempre y cuando esta tarea haya sido compartida previamente
3. Un usuario puede modificar cualquier campo de una tarea que le hayan compartido, excepto la de *Creado por*
4. Un usuario puede ver toda la lista de tareas públicas, así como editarlas, completarlas y eliminarlas

5. Los métodos GET deben funcionar para fines de paginación en el Frontend y mostrar cuántos resultados encontró en cada llamada en la misma respuesta
6. Búsqueda de tareas:
  - a. Un usuario admin puede realizar una búsqueda de tareas
  - b. Las tareas se ponderan según el match de sus campos con lo siguiente:
    - i. Palabra clave
    - ii. Estatus de compleción
    - iii. ¿Es pública?
    - iv. Cantidad de usuarios compartidos
    - v. Días restantes para vencimiento
    - vi. Formato de archivo

#### Aspectos a evaluar:

1. Funcionamiento de la API REST
2. Utilización de framework para API REST
3. Estructura del proyecto
4. Documentación del código y endpoints
5. Buenas prácticas de programación

#### Se sugiere utilizar el siguiente stack de herramientas:

1. Node.JS
2. Express.JS
3. MySQL