

Atividade Prática 01 - NODE

- Exercício pode ser feito individual ou em dupla
- Códigos desenvolvidos por Inteligência Artificial (IA) ou plágios acarretam nota zero.

Desenvolva uma API para realizar o CRUD de suas *tasks* (tarefas).

A API deve conter as seguintes funcionalidades:

- Criação de uma task
- Listagem de todas as tasks
- Atualização de uma task pelo `id`
- Remover uma task pelo `id`
- Marcar pelo `id` uma task como completa

Rotas e regras de negócio

Antes das rotas, vamos entender qual a estrutura (propriedades) que uma task deve ter:

- `id` - Identificador único de cada task
- `title` - Título da task
- `description` - Descrição detalhada da task
- `completed_at` - Data de quando a task foi concluída. O valor inicial deve ser `null`
- `created_at` - Data de quando a task foi criada.
- `updated_at` - Deve ser sempre alterado para a data de quando a task foi atualizada.

Rotas:

- POST - `/tasks`

Deve ser possível criar uma task no banco de dados, enviando os campos `title` e `description` por meio do `body` da requisição.

Ao criar uma task, os campos: `id`, `created_at`, `updated_at` e `completed_at` devem ser preenchidos automaticamente, conforme a orientação das propriedades acima.

- GET - `/tasks`

Deve ser possível listar todas as tasks salvas no banco de dados.

Também deve ser possível realizar uma busca, filtrando as tasks pelo `title` e `description`

- PUT - /tasks/:id

Deve ser possível atualizar uma task pelo `id`.

No `body` da requisição, deve receber somente o `title` e/ou `description` para serem atualizados.

Se for enviado somente o `title`, significa que o `description` não pode ser atualizado e vice-versa.

Antes de realizar a atualização, deve ser feito uma validação se o `id` pertence a uma task salva no banco de dados.

- DELETE - /tasks/:id

Deve ser possível remover uma task pelo `id`.

Antes de realizar a remoção, deve ser feito uma validação se o `id` pertence a uma task salva no banco de dados.

- PATCH - /tasks/:id/complete

Deve ser possível marcar a task como completa ou não. Isso significa que se a task estiver concluída, deve voltar ao seu estado “normal”.

Antes da alteração, deve ser feito uma validação se o `id` pertence a uma task salva no banco de dados.

Algumas sugestões a serem consideradas para ser implementado:

- Validar se as propriedades `title` e `description` das rotas POST e PUT estão presentes no `body` da requisição.
- Nas rotas que recebem o `/:id`, além de validar se o `id` existe no banco de dados, retornar à requisição com uma mensagem informando que o registro não existe.