

API de Gestão de Tarefas com Autenticação JWT

Objetivo

Desenvolver uma API RESTful para gestão de tarefas, com autenticação de usuários utilizando JWT, validação de login com senha em texto plano, e persistência dos dados em um banco de dados PostgreSQL via Supabase, utilizando Sequelize como ORM.

Requisitos Funcionais

Usuários

- Cadastrar novo usuário: POST /usuarios
- Fazer login com email e senha (texto plano): POST /login
- Receber token JWT válido após login

Tarefas

- Criar nova tarefa: POST /tarefas (usuário autenticado)
- Listar tarefas do usuário logado: GET /tarefas
- Atualizar tarefa: PUT /tarefas/:id
- Remover tarefa: DELETE /tarefas/:id

Requisitos de Segurança

- Usar JWT para proteger todas as rotas relacionadas a tarefas
- Validar token JWT via middleware (Authorization: Bearer TOKEN)
- Garantir que o usuário só possa acessar suas próprias tarefas

Estrutura do Banco de Dados (PostgreSQL - Supabase)

Tabela: Usuario

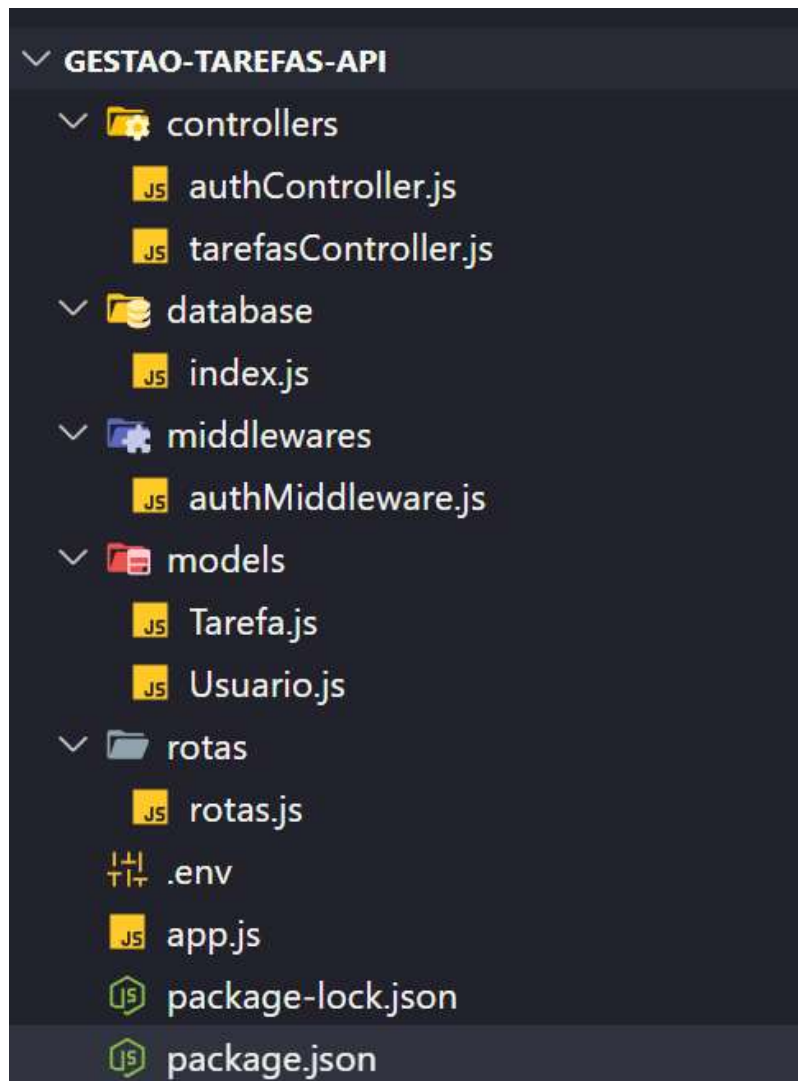
Campo	Tipo	Descrição
id	INTEGER PK	Identificador do usuário
nome	VARCHAR	Nome do usuário
email	VARCHAR	Email (único)
senha	VARCHAR	Senha em texto plano

Tabela: Tarefa

Campo	Tipo	Descrição
id	INTEGER PK	Identificador da tarefa
titulo	VARCHAR	Título da tarefa
descricao	TEXT	Detalhes da tarefa
status	VARCHAR	Status da tarefa (ex: pendente)
usuario_id	INTEGER (FK)	FK referenciando a tabela Usuario

Configure o arquivo .env

```
PORT=3000
BANCO_DE_DADOS=postgres://usuario:senha@db.supabase.co:5432/nome_do_banco
SEGREDO_JWT=sua_chave_secreta
```



Testes com o Postman

1. Cadastrar um novo usuário

Método: POST

URL: `http://localhost:3000/usuarios`

Body (raw → JSON):

```
{
  "nome": "Maria Silva",
  "email": "maria@email.com",
  "senha": "123456"
}
```

2. Login do usuário

Método: POST

URL: `http://localhost:3000/login`

Body (raw → JSON):

```
{
  "email": "maria@email.com",
  "senha": "123456"
}
```

A resposta trará um objeto com o token JWT. Copie esse token, pois ele será usado nos próximos testes.

3. Criar uma nova tarefa

Método: POST

URL: `http://localhost:3000/tarefas`

Headers:

`Authorization: Bearer SEU_TOKEN_JWT`

Body (raw → JSON):

```
{
  "titulo": "Estudar Node.js",
  "descricao": "Fazer o exercício de Sequelize",
  "status": "pendente"
}
```

A tarefa será criada vinculada ao usuário logado.

4. Listar todas as tarefas do usuário logado

Método: GET

URL: `http://localhost:3000/tarefas`

Headers:

`Authorization: Bearer SEU_TOKEN_JWT`

Retorna um array com todas as tarefas do usuário autenticado.

5. Atualizar uma tarefa

Método: PUT

URL: `http://localhost:3000/tarefas/ID_DA_TAREFA`

Headers:

`Authorization: Bearer SEU_TOKEN_JWT`

Body (raw → JSON):

```
{
  "titulo": "Estudar Node.js e JWT",
  "descricao": "Completar os testes com o Postman",
  "status": "concluida"
}
```

6. Excluir uma tarefa

Método: DELETE

URL: `http://localhost:3000/tarefas/ID_DA_TAREFA`

Headers:

`Authorization: Bearer SEU_TOKEN_JWT`