Consigna: Servidor con Express + MongoDB + Autenticación (JWT) + Arquitectura MVC

Descripción

Crear un servidor backend utilizando **Express** y **MongoDB** que implemente autenticación con **JWT** y siga el patrón de arquitectura **MVC (Modelo–Vista–Controlador)**.
El proyecto deberá incluir al menos una entidad protegida (por ejemplo, `Producto`, `Evento` o `Tarea`) asociada al usuario autenticado.

Objetivos

- Comprender la integración entre Express y MongoDB a través de Mongoose.
- Aplicar buenas prácticas de arquitectura modular (MVC).
- Implementar autenticación segura con **bcrypt** y **JWT**.
- Desarrollar endpoints públicos y privados con control de acceso mediante middleware.
- Manejar errores de manera centralizada.

Requisitos Técnicos

1. **Servidor Express** con configuración base y middlewares (`cors`, `express.json`, etc.).
2. **Conexión a MongoDB** (local o Atlas) usando Mongoose.
3. **Autenticación JWT** con endpoints para registro y login de usuarios.
4. **Rutas protegidas** que requieran token válido.
5. **Arquitectura MVC** correctamente implementada: separación entre rutas, controladores, servicios/modelos.
6. **Variables de entorno** gestionadas con `.env`.
7. **Manejo de errores** mediante middleware global.
8. **Documentación** en un archivo `README.md` con instrucciones para ejecutar el proyecto.

Endpoints mínimos

Autenticación (públicos)

- `POST /api/auth/register` → Crea un nuevo usuario y devuelve token.
- `POST /api/auth/login` → Valida credenciales y devuelve token.

Entidad protegida (privados)

- `GET /api/[entidad]` → Lista elementos del usuario autenticado.

- `POST /api/[entidad]` → Crea un nuevo elemento asociado al usuario.
- `PATCH /api/[entidad]/:id` → Actualiza si pertenece al usuario.
- `DELETE /api/[entidad]/:id` → Elimina si pertenece al usuario.

Entregables

- Carpeta del proyecto con estructura MVC.
- Archivo `.env.example` con variables de entorno necesarias.
- Archivo `README.md` con:
  - Descripción del proyecto.
  - Instrucciones de instalación y ejecución.
  - Ejemplos de requests.
- Colección de pruebas (Postman o Thunder Client).
- (Opcional) Deploy funcional en Render o Railway.

Criterios de evaluación

- **Organización (25%)**: uso correcto de la arquitectura MVC.
- **Seguridad (25%)**: manejo de contraseñas y tokens JWT.
- **Funcionalidad (25%)**: endpoints operativos y protección por usuario.
- **Calidad del código (15%)**: claridad, validaciones y mensajes coherentes.
- **Documentación (10%)**: README completo, ejemplos y scripts.