

Proyecto: API de Gestión de Proyectos y Tareas

Descripción General

Desarrollar una API RESTful completa para la gestión de proyectos y tareas, donde los usuarios puedan crear, organizar y colaborar en proyectos con diferentes niveles de acceso y funcionalidades inteligentes.

Modelos de Datos Requeridos

1. Usuarios (Users)

- `_id`: ObjectId
- `firstName`: String
- `lastName`: String
- `email`: String (único)
- `password`: String (encriptada)
- `avatar`: String (URL)
- `phone`: String
- `globalRole`: ObjectId (referencia a Role)
- `isActive`: Boolean
- `isEmailVerified`: Boolean
- `lastLogin`: Date
- `createdAt`: Date
- `updatedAt`: Date

2. Roles (Roles)

- `_id`: ObjectId
- `name`: String (Admin, Project Manager, Developer, Viewer)
- `description`: String
- `isActive`: Boolean
- `createdAt`: Date
- `updatedAt`: Date

3. Categorías (Categories)

- `_id`: ObjectId
- `name`: String
- `description`: String
- `isActive`: Boolean
- `createdBy`: ObjectId (referencia a User)

- createdAt: Date
- updatedAt: Date

4. Proyectos (Projects)

- _id: ObjectId
- name: String
- description: String
- category: ObjectId (referencia a Category)
- owner: ObjectId (referencia a User)
- members: Array de Objects { user: ObjectId, role: ObjectId, joinedAt: Date }
- status: ObjectId (referencia a State)
- priority: String (Low, Medium, High, Critical)
- startDate: Date
- endDate: Date
- estimatedHours: Number
- actualHours: Number
- budget: Number
- isActive: Boolean
- tags: Array de Strings
- createdAt: Date
- updatedAt: Date

5. Tareas (Tasks)

- _id: ObjectId
- title: String
- description: String
- project: ObjectId (referencia a Project)
- assignedTo: ObjectId (referencia a User)
- createdBy: ObjectId (referencia a User)
- status: ObjectId (referencia a State)
- priority: String (Low, Medium, High, Critical)
- estimatedHours: Number
- actualHours: Number
- startDate: Date
- dueDate: Date
- completedAt: Date
- tags: Array de Strings
- isActive: Boolean
- createdAt: Date
- updatedAt: Date

6. Comentarios (Comments) sobre los proyectos

- `_id`: ObjectId
- `content`: String
- `author`: ObjectId (referencia a User)
- `projectid`: (referencia a Projects)
 - `editedAt`: Date
 - `createdAt`: Date
 - `updatedAt`: Date

7. Estados (States)

- `_id`: ObjectId
- `name`: String
- `description`: String
- `type`: String (Project, Task)
- `isActive`: Boolean
- `createdAt`: Date
- `updatedAt`: Date

Funcionalidades con IA

- **Generación automática de tareas**: Basada en descripción del proyecto
- **Sugerencias de mejora**: Análisis de comentarios y estados
- **Estimación de tiempo**: Predicción de duración de tareas
- **Resúmenes de progreso**: Reportes automáticos del estado del proyecto

Flujo de Información y Funcionamiento de la API

Descripción General del Flujo

La API sigue un flujo estructurado donde los usuarios interactúan con proyectos y tareas a través de diferentes niveles de autorización y estados controlados.

Flujo Principal de la Aplicación

1. Registro y Autenticación

El usuario se registra proporcionando información básica y recibe un rol por defecto. Al iniciar sesión, obtiene un token JWT que debe incluir en todas las peticiones autenticadas. El token contiene información del usuario y sus permisos.

2. Gestión de Usuarios y Roles

Los administradores pueden asignar roles específicos a los usuarios, lo que determina qué acciones pueden realizar en el sistema. Cada rol tiene permisos definidos que se validan en cada endpoint.

3. Creación de Categorías

Las categorías se crean para clasificar proyectos (ej: "Desarrollo Web", "Mobile", "DevOps"). Estas sirven como filtros y organizadores visuales.

4. Ciclo de Vida de un Proyecto

- **Creación:** Un usuario con permisos crea un proyecto, asignándole una categoría y estado inicial
- **Configuración de Equipo:** El owner del proyecto invita miembros y les asigna roles específicos dentro del proyecto
- **Gestión de Estados:** El proyecto transita por diferentes estados según el progreso (Planificación → En Progreso → En Revisión → Completado)

5. Gestión de Tareas

- **Creación:** Dentro de un proyecto, se crean tareas específicas
- **Asignación:** Las tareas se asignan a miembros del equipo
- **Seguimiento:** Las tareas cambian de estado según el progreso
- **Dependencias:** Las tareas pueden depender de otras, creando un flujo de trabajo estructurado

6. Sistema de Comentarios

Los comentarios permiten comunicación en tiempo real sobre los proyectos.

7. Control de Estados

Los estados controlan el flujo de trabajo tanto para proyectos como para tareas, asegurando que sigan una progresión lógica y permitiendo transiciones válidas entre diferentes fases del trabajo.

Flujo de Datos entre Modelos

Relaciones Clave

- **Usuario → Proyecto:** Un usuario puede ser owner o miembro de múltiples proyectos

- **Proyecto → Tareas:** Un proyecto contiene múltiples tareas
- **Usuario → Tareas:** Un usuario puede tener múltiples tareas asignadas
- **Comentarios:** Se relacionan con proyectos específicos
- **Estados:** Controlan el flujo de proyectos y tareas por separado

Validaciones de Flujo

- Solo usuarios con permisos pueden crear proyectos
- Solo miembros del proyecto pueden ver/editar sus tareas

Integración con Inteligencia Artificial

Flujo de IA

1. **Entrada:** El usuario proporciona descripción del proyecto o contexto
2. **Procesamiento:** La API envía la información a servicios de IA (ChatGPT/Gemini)
3. **Análisis:** La IA procesa la información y genera sugerencias
4. **Respuesta:** Los resultados se formatean y devuelven al usuario
5. **Aplicación:** El usuario puede aplicar las sugerencias generadas

Casos de Uso de IA

- **Generación de Tareas:** Crear automáticamente tareas basadas en descripción del proyecto
- **Análisis de Progreso:** Evaluar el estado del proyecto y sugerir mejoras
- **Estimación de Tiempo:** Predecir duración de tareas basándose en descripción y contexto
- **Resúmenes Inteligentes:** Generar reportes automáticos del progreso

Flujo de Permisos y Seguridad

Niveles de Autorización

1. **Global:** Determinado por el rol del usuario en el sistema
2. **Proyecto:** Rol específico del usuario dentro del proyecto
3. **Recurso:** Permisos específicos sobre tareas/comentarios propios

Validación de Permisos

Cada endpoint valida permisos en este orden:

1. Autenticación (token JWT válido)
2. Autorización global (rol del usuario)

3. Autorización específica (permisos sobre el recurso)

Endpoints de la API

Autenticación

- POST /api/auth/register - Registro de usuario
- POST /api/auth/login - Inicio de sesión
- POST /api/auth/refresh - Renovar token JWT
- POST /api/auth/logout - Cerrar sesión
- POST /api/auth/forgot-password - Recuperar contraseña
- POST /api/auth/reset-password - Restablecer contraseña

Usuarios

- GET /api/users - Listar usuarios (Admin)
- GET /api/users/profile - Perfil del usuario actual
- PUT /api/users/profile - Actualizar perfil
- DELETE /api/users/:id - Eliminar usuario (Admin)
- PUT /api/users/:id/role - Cambiar rol (Admin)

Roles

- GET /api/roles - Listar roles disponibles
- POST /api/roles - Crear nuevo rol (Admin)
- PUT /api/roles/:id - Actualizar rol (Admin)
- DELETE /api/roles/:id - Eliminar rol (Admin)

Categorías

- GET /api/categories - Listar categorías
- POST /api/categories - Crear categoría
- PUT /api/categories/:id - Actualizar categoría
- DELETE /api/categories/:id - Eliminar categoría

Proyectos

- GET /api/projects - Listar proyectos del usuario
- POST /api/projects - Crear proyecto
- GET /api/projects/:id - Obtener proyecto específico
- PUT /api/projects/:id - Actualizar proyecto
- DELETE /api/projects/:id - Eliminar proyecto
- POST /api/projects/:id/members - Agregar miembro al proyecto

- DELETE /api/projects/:id/members/:userId - Remover miembro
- PUT /api/projects/:id/status - Cambiar estado del proyecto

Tareas

- GET /api/projects/:projectId/tasks - Listar tareas del proyecto
- POST /api/projects/:projectId/tasks - Crear tarea
- GET /api/tasks/:id - Obtener tarea específica
- PUT /api/tasks/:id - Actualizar tarea
- DELETE /api/tasks/:id - Eliminar tarea
- PUT /api/tasks/:id/status - Cambiar estado de tarea
- PUT /api/tasks/:id/assign - Asignar tarea a usuario
- GET /api/tasks/my-tasks - Tareas asignadas al usuario X

Comentarios

- GET /api/projects/:id/comments - Comentarios del proyecto
- POST /api/projects/:id/comments - Comentar en proyecto
- PUT /api/comments/:id - Editar comentario
- DELETE /api/comments/:id - Eliminar comentario

Estados

- GET /api/states/projects - Estados disponibles para proyectos
- GET /api/states/task - Estados disponibles para tareas
- POST /api/states - Crear nuevo estado (Admin)
- PUT /api/states/:id - Actualizar estado (Admin)

Inteligencia Artificial

- POST /api/ai/generate-tasks - Generar tareas automáticamente basadas en descripción del proyecto
- POST /api/ai/analyze-project - Análisis y sugerencias de mejora para el proyecto
- POST /api/ai/estimate-time - Estimación de tiempo para completar tareas
- POST /api/ai/generate-summary - Generar resumen del progreso del proyecto
- POST /api/ai/suggest-improvements - Sugerencias basadas en comentarios y estados

Requisitos Técnicos

Validaciones

- Validar todos los inputs con express-validator
- Validar permisos según roles

- Validar estados válidos para transiciones

Seguridad

- Autenticación JWT
- Encriptación de contraseñas con bcryptjs
- Middleware de autorización por roles
- Validación de propiedad de recursos

Base de Datos

- MongoDB con Mongoose
- Relaciones entre modelos
- Índices para optimización
- Validaciones a nivel de esquema

Funcionalidades Adicionales

- Paginación en listados
- Filtros y búsqueda
- Ordenamiento de resultados
- Manejo de errores centralizado
- Logging de actividades
- Rate limiting

Nivel de Complejidad

Este proyecto está diseñado para nivel intermedio, incluyendo:

- Arquitectura MVC
- Middleware personalizado
- Relaciones complejas entre modelos
- Integración con APIs externas (IA)
- Manejo de roles y permisos
- Validaciones avanzadas
- Optimización de consultas