

# ANEXO 01 – MANUAL TÉCNICO

## 1. Información General del Sistema

**Nombre del sistema:** AI Research Assistant MERN

**Versión:** 1.0

**Arquitectura:** MERN (MongoDB – Express – React – Node.js) + APIs de IA + n8n

**Base de datos:** MongoDB Atlas

---

## 2. Requerimientos Técnicos

### 2.1 Requerimientos de Software

- Node.js v18+
- MongoDB Atlas / Local Mongo v6+
- npm v9+
- React v18+
- Express v4+
- Docker (Opcional)
- n8n (Para automatizaciones)

### 2.2 Requerimientos de Hardware

- CPU mínimo: 2 núcleos
  - RAM: 8 GB mínimo
  - Espacio en disco: 10 GB
  - Conexión estable a Internet
- 

## 3. Arquitectura del Sistema

El sistema sigue una arquitectura **Cliente-Servidor**, donde:

- **Frontend React** presenta la interfaz de usuario.
- **Backend Node/Express** expone endpoints API REST.
- **MongoDB** almacena información de usuarios, proyectos, documentos y resúmenes.
- **Servicios de IA** permiten generar resúmenes y sugerencias.
- **n8n** gestiona automatizaciones.

Diagrama conceptual de arquitectura:

React (Frontend) → Express/Node (API) → MongoDB Atlas
---

↓  
APIs de IA externas  
↓  
n8n

## 4. Instalación Técnica

### 4.1 Instalación del backend

```
cd backend  
npm install  
npm start
```

### 4.2 Instalación del Frontend

```
cd frontend  
npm install  
npm start
```

### 4.3 Variables de entorno

Crear archivo .env en backend:

```
MONGODB_URI=xxx  
JWT_SECRET=xxx  
OPENAI_KEY=xxx\```\n\n---\n\n## 5. Base de Datos  
Las principales colecciones en MongoDB son:  
- users  
- projects  
- documents  
- summaries  
- suggestions  
- notifications  
- auditLogs
```

Ejemplo documento `project`:

```
{_id: xxx, ownerId: xxx, title:"Proyecto IA", description:"automatización académica",  
createdAt: Date }
```

```
---\n\n## 6. Endpoints Principales API REST
```

Método	Endpoint	Descripción
POST	/auth/login	Inicio de sesión
POST	/auth/register	Registro de usuarios
GET	/projects	Lista proyectos
POST	/projects	Crear proyecto
POST	/summary	Generar resumen
GET	/documents	Consultar documentos

---

## ## 7. Seguridad

- Cifrado SHA-256 en contraseñas
- JWT en autenticación
- Gestión de roles
- Prevención de inyección NoSQL

---

## ## 8. Pruebas Técnicas

### Pruebas unitarias (Jest + Supertest)  
Ejecución:

npm run test

### Cobertura de código  
de backend y frontend:

- Backend >= 85%
- Frontend >= 80%

---

## ## 9. Automatización (n8n)

Usado para:

- búsqueda de artículos periódicos
- notificación automática
- workflows de procesamiento

Ejemplo de flujo:

Evento → consulta Scholar → filtra resultados → notifica usuario ``

---

## 10. Mantenimiento y soporte

**Responsables:** - Administrador del sistema - Equipo de TI

**Tareas periódicas:** - Backup MongoDB cada 7 días - Verificación integridad JWT - Depuración de logs > 90 días

---

## 11. Respaldo y recuperación de datos

- Respaldos automáticos de MongoDB
  - Restauración desde snapshot
- 

## 12. Versionamiento del código

Repositorio en GitHub con: - ramas main, dev - commits con convención estándar - revisión por PR

---