

# ■ Requerimientos y Estructura del Sistema Académico

Este documento describe los requerimientos funcionales, la estructura de datos y la arquitectura general del Sistema Académico. El sistema permite administrar estudiantes, profesores, cursos, inscripciones y notas, garantizando una gestión eficiente de la información educativa.

## 1. Requerimientos Funcionales

Módulo	Descripción
Estudiantes	Permite registrar, actualizar, consultar y eliminar información de los estudiantes.
Profesores	Administra la información personal y académica de los profesores.
Cursos	Gestiona la creación, edición y asignación de cursos a profesores.
Inscripciones	Registra las inscripciones de los estudiantes en los cursos disponibles.
Notas	Permite registrar las calificaciones asociadas a las inscripciones de los estudiantes.

## 2. Requerimientos No Funcionales

- **Lenguaje backend:** C# (.NET 8 Web API). - **Base de datos:** SQLite (en desarrollo) o PostgreSQL (en producción). - **Frontend:** React + Vite + Axios. - **ORM:** Entity Framework Core. - **Autenticación:** JWT. - **Interfaz de usuario:** Minimalista, moderna y responsiva. - **Documentación de API:** Swagger (integrado en .NET).

## 3. Estructura del Sistema

El proyecto está dividido en dos partes principales: - **Backend (.NET API):** contiene los modelos, controladores, servicios y contexto de base de datos. - **Frontend (React):** gestiona la interfaz de usuario, conectándose al API a través de servicios Axios.

Estructura general:

Carpeta	Contenido principal
backend/NotesApi/Models	Clases del modelo de base de datos (Estudiante, Profesor, Curso, etc.)
backend/NotesApi/Controllers	Controladores API (Estudiantes, Cursos, Notas, etc.)
backend/NotesApi/Services	Lógica de negocio y validaciones.
backend/NotesApi/Data	Contexto de base de datos (AppDbContext.cs).
frontend/src/pages	Páginas de React: Students, Courses, Enrollments, Grades.
frontend/src/services	Servicios Axios para comunicación con el backend.

## 4. Diagrama Lógico de Base de Datos

El modelo relacional se basa en las siguientes tablas principales: - \*\*estudiantes(id, nombre, apellido, email, fecha\_nacimiento)\*\* - \*\*profesores(id, nombre, apellido, email, departamento)\*\* - \*\*cursos(id, nombre, codigo, creditos, profesor\_id)\*\* - \*\*inscripciones(id, estudiante\_id, curso\_id, fecha\_inscripcion)\*\* - \*\*notas(id, inscripcion\_id, nota, fecha\_registro)\*\*

## 5. Conclusión

El sistema académico propuesto permite una gestión completa del proceso educativo: desde la administración de usuarios (estudiantes y profesores), hasta la gestión de cursos, inscripciones y notas, con arquitectura moderna, escalable y adaptable a entornos web o locales.