# Prueba Técnica: Sistema de Agendamiento de Clases para un Colegio

**Objetivo**

Desarrollar una interfaz de administrador que permita gestionar el agendamiento de clases en un colegio. La interfaz debe permitir la creación y gestión de profesores, estudiantes y clases. Cada clase puede tener un único profesor y múltiples estudiantes asignados. El objetivo principal es implementar el CRUD para profesores, estudiantes y clases, así como la asignación de profesores y estudiantes a las clases.

Además, desarrollar el frontend utilizando React y la librería de componentes Material-UI para garantizar una buena usabilidad y diseño.

**Requerimientos Funcionales**

1. Gestión de Profesores:
    - Crear un nuevo profesor.
    - Editar los detalles de un profesor existente.
    - Eliminar un profesor.
    - Listar todos los profesores.
2. Gestión de Estudiantes:
    - Crear un nuevo estudiante.
    - Editar los detalles de un estudiante existente.
    - Eliminar un estudiante.
    - Listar todos los estudiantes.
3. Gestión de Clases:
    - Crear una nueva clase.
    - Editar los detalles de una clase existente.
    - Eliminar una clase.
    - Listar todas las clases.
    - Asignar un profesor a una clase.
    - Asignar estudiantes a una clase.
    - Visualizar los estudiantes y el profesor asignados a una clase.
4. Requerimientos de Usabilidad:
    - Utilizar la librería Material-UI para el desarrollo del frontend.
    - Asegurarse de que la interfaz sea intuitiva y fácil de usar.
    - No se requiere un diseño específico, pero se valorará positivamente una buena usabilidad y diseño.

**Detalles Técnicos**

1. Backend:
    - Utilizar NestJS como framework backend.
    - Utilizar MySQL como base de datos.
    - Utilizar TypeORM para la gestión de las entidades y sus relaciones.
2. Frontend:
    - Utilizar React para el desarrollo del frontend.
    - Utilizar la librería Material-UI para los componentes de la interfaz de usuario (https://mui.com/material-ui/).
3. Entidades:
    - Profesor (Teacher):
        - ID (número entero, autogenerado)
        - Nombre (cadena de texto)
        - Apellido (cadena de texto)
        - Email (cadena de texto)
    - Estudiante (Student):
        - ID (número entero, autogenerado)
        - Nombre (cadena de texto)
        - Apellido (cadena de texto)
        - Email (cadena de texto)
    - Clase (Class):
        - ID (número entero, autogenerado)
        - Nombre de la clase (cadena de texto)
        - Descripción (cadena de texto)
        - Profesor (relación uno a muchos con Profesor)
        - Estudiantes (relación muchos a muchos con Estudiante)
4. API Endpoints:
    - Profesores:
        - `POST /teachers` - Crear un nuevo profesor.
        - `GET /teachers` - Listar todos los profesores.
        - `GET /teachers/:id` - Obtener los detalles de un profesor específico.
        - `PUT /teachers/:id` - Actualizar los detalles de un profesor.
        - `DELETE /teachers/:id` - Eliminar un profesor.
    - Estudiantes:
        - `POST /students` - Crear un nuevo estudiante.
        - `GET /students` - Listar todos los estudiantes.
        - `GET /students/:id` - Obtener los detalles de un estudiante específico.
        - `PUT /students/:id` - Actualizar los detalles de un estudiante.

        - `DELETE /students/:id` - Eliminar un estudiante.
    - Clases:
        - `POST /classes` - Crear una nueva clase.
        - `GET /classes` - Listar todas las clases.
        - `GET /classes/:id` - Obtener los detalles de una clase específica.
        - `PUT /classes/:id` - Actualizar los detalles de una clase.
        - `DELETE /classes/:id` - Eliminar una clase.
        - `POST /classes/:id/assign-teacher` - Asignar un profesor a una clase.
        - `POST /classes/:id/assign-students` - Asignar estudiantes a una clase.
        - `GET /classes/:id/students` - Obtener la lista de estudiantes asignados a una clase.

**Implementación Sugerida**

1. Configuración del Proyecto:
    - Configurar un nuevo proyecto NestJS y MySQL.
    - Instalar y configurar TypeORM.
    - Configurar un nuevo proyecto React.
    - Instalar Material-UI en el proyecto React.
2. Entidades y Repositorios en Backend:
    - Crear las entidades `Teacher`, `Student`, y `Class`.
    - Configurar las relaciones entre las entidades.
3. Servicios y Controladores en Backend:
    - Implementar servicios para manejar la lógica de negocio.
    - Crear controladores para gestionar los endpoints API.
4. Interfaz de Usuario en Frontend:
    - Crear componentes para las vistas de profesores, estudiantes y clases.
    - Utilizar Material-UI para los componentes de la interfaz.
    - Implementar formularios para crear y editar profesores, estudiantes y clases.
    - Implementar tablas o listas para mostrar los datos.
    - Asegurarse de que la asignación de profesores y estudiantes a clases sea intuitiva.

## Consideraciones Adicionales

1. **Código limpio y bien documentado:** Asegúrese de que su código esté limpio, bien estructurado y debidamente documentado. Esto incluye comentarios en el código donde sea necesario, documentación de las funciones y clases, así como un README completo que explique cómo configurar y ejecutar el proyecto.
2. **Uso de IA como apoyo:** Se permite y se alienta el uso de herramientas de inteligencia artificial como ChatGPT o GitHub Copilot para ayudar en el desarrollo. Estas herramientas pueden ser útiles para resolver dudas, generar fragmentos de código y mejorar la eficiencia en la programación. Se bonificará si se comparten los prompts usados en el proyecto.
3. **Proporcionar la URL del repositorio GIT:** Al finalizar el proyecto, proporcione la URL del repositorio GIT donde se encuentra el código fuente. Asegúrese de que el repositorio sea accesible y contenga toda la información necesaria para evaluar su trabajo.
4. **Valoración adicional con FIGMA:** Si decide agregar un diseño de la interfaz en FIGMA, proporcione el enlace al proyecto de FIGMA junto con su repositorio GIT. La inclusión de un diseño detallado y bien pensado de la interfaz de usuario será considerado un valor añadido y se reflejará positivamente en la evaluación final.

Mucha Suerte!