

¡Por supuesto! Aquí tienes la **explicación técnica y detallada** de todo el sistema **EduSphere**. Está diseñada para que puedas explicarla en tu exposición demostrando dominio total de la lógica y el código.

Arquitectura del Sistema EduSphere

EduSphere es una aplicación web **Full-Stack** que sigue la arquitectura **Cliente-Servidor**. No es un simple sitio web, es una plataforma dinámica donde el Frontend y el Backend se comunican constantemente.

1. Tecnologías Utilizadas (El "Stack")

Para construir esto, utilizamos un stack moderno y profesional conocido como **PERN** (Postgres, Express, React, Node), potenciado con IA.

- **Frontend (Lo que ve el usuario):**
 - **Lenguaje:** JavaScript (JSX).
 - **Framework:** **React** (versión 18/19) ejecutado con **Vite** para máxima velocidad.
 - **Estilos:** **TailwindCSS** (para un diseño moderno y responsivo sin escribir cientos de líneas de CSS manual).
 - **Comunicación:** **Axios** (para conectar con el Backend).
 - **Backend (El cerebro):**
 - **Entorno:** **Node.js**.
 - **Framework:** **Express.js** (para manejar las rutas y el servidor HTTP).
 - **Seguridad:** **Bcrypt** (para encriptar contraseñas) y **JWT** (JSON Web Tokens para mantener la sesión iniciada).
 - **Base de Datos (La memoria):**
 - **Motor:** **PostgreSQL** (Base de datos relacional robusta).
 - **ORM:** **Prisma** (Herramienta que nos permite "hablar" con la base de datos usando código JavaScript en lugar de SQL puro).
 - **Inteligencia Artificial:**
 - **API:** **OpenAI API** (Modelo GPT-3.5 Turbo).
-

2. Lógica del Sistema (¿Cómo funciona todo junto?)

El sistema se divide en módulos lógicos que interactúan entre sí:

A. Módulo de Seguridad (Autenticación Blindada)

No usamos un login simple. Implementamos **2FA (Autenticación de Doble Factor)** obligatoria.

1. **El Usuario:** Ingresa correo y contraseña.
2. **El Backend:** Verifica la contraseña (encriptada). Si es correcta, genera un código numérico (OTP) aleatorio y lo guarda en la Base de Datos con una fecha de expiración (10 mins).
3. **Nodemailer:** Envía ese código al correo real del usuario.
4. **Verificación:** El usuario ingresa el código. El backend verifica que coincida y no haya expirado. Solo entonces entrega el **Token JWT** (la "llave digital") para usar el sistema.

B. Módulo Académico (Exámenes y Materiales)

- **Materiales:** Usamos la librería **Multer** para recibir archivos. Cuando un profesor sube un PDF, el archivo se guarda en una carpeta local del servidor (/uploads), y en la base de datos solo guardamos la "dirección" (URL) de donde está ese archivo.
- **Exámenes:** Los exámenes no son estáticos. Tienen un campo assignedTo (array de IDs) que decide quién puede verlos. Si está vacío, es público. Si tiene IDs, es privado. Además, guardamos un JSON con las preguntas y respuestas correctas, pero el Backend **filtra** las respuestas correctas antes de enviarlas al estudiante para evitar trampas.

3. El Chatbot IA: ¿Cómo lo hiciste? (La Magia Explicada) 🧠

Esta es la parte diferenciadora ("WAOOO"). No es un chat genérico como ChatGPT; es un **Asistente Contextual (RAG Básico)**.

El Problema: ChatGPT normal no sabe qué PDF subió tu profesor.

Tu Solución: Le "inyectamos" el conocimiento antes de que responda.

Paso a Paso Técnico:

1. **Selección:** En el Frontend (StudentAssistantPage.jsx), el estudiante marca con checkboxes qué materiales quiere estudiar.
2. **Petición:** El Frontend envía al Backend:
 - La pregunta del usuario.
 - Los IDs de los materiales seleccionados ([id1, id2]).
3. **Recuperación (Backend):**
 - El controlador assistant.controller.js recibe los IDs.
 - Usa **Prisma** para buscar en la Base de Datos el **texto/contenido** de esos materiales específicos.
4. **Ingeniería de Prompts (El Truco):**
 - El Backend construye un mensaje oculto para la IA (Prompt de Sistema) que dice algo así: *"Eres un tutor experto. Basa tu respuesta ÚNICAMENTE en el siguiente contexto que te proporciono: [Aquí pegamos el texto de los materiales de la Base de Datos]. Pregunta del usuario: [Pregunta]"*.
5. **Respuesta:** OpenAI lee el contexto y responde. El estudiante siente que la IA "leyó" su documento.

Flashcards (Tarjetas de Estudio):

Usamos la misma lógica, pero en el Prompt le ordenamos a la IA: "Devuélveme la respuesta en formato JSON estricto con pares de pregunta/respuesta". Así el Frontend puede tomar ese JSON y dibujar las tarjetas giratorias automáticamente.

Resumen para tu Exposición:

"EduSphere combina la robustez de una base de datos relacional (PostgreSQL) para la gestión académica segura, con la flexibilidad de la Inteligencia Artificial (OpenAI). No solo digitalizamos exámenes, sino que transformamos materiales estáticos en tutores interactivos inyectando el contenido de los cursos directamente en el contexto de la IA en tiempo real."

Concepto General

EduSphere es una **Plataforma Integral de Evaluación y Aprendizaje Asistido por IA**. No es solo un sistema para subir tareas (como un LMS tradicional), sino un ecosistema inteligente diseñado para transformar la educación mediante herramientas avanzadas de **Inteligencia Artificial**, ofreciendo aprendizaje personalizado y automatización para docentes y estudiantes.

Objetivos y Misión

La misión principal del proyecto es **democratizar el acceso a la educación de calidad** utilizando el poder de la IA. Sus objetivos específicos son:

- **Personalización:** Adaptar el contenido al ritmo y estilo de cada estudiante.
- **Eficiencia Docente:** Ahorrar tiempo a los profesores automatizando tareas repetitivas como la calificación y generación de feedback.
- **Estudio Activo:** Convertir materiales estáticos (PDFs) en herramientas interactivas de estudio (Chatbots y Flashcards).
- **Seguridad:** Garantizar la protección de datos mediante autenticación robusta de doble factor.

Funcionalidades Clave (¿Qué hace?)

1. **Seguridad de Nivel Bancario:**
 - Implementa login con **Autenticación de Dos Pasos (2FA)**. Al ingresar la contraseña, el sistema envía un código OTP al correo del usuario para verificar su identidad antes de dar acceso.
2. **Gestión Académica Completa:**
 - **Docentes:** Pueden crear exámenes con preguntas dinámicas, definir tiempos límite, asignar exámenes a estudiantes específicos (o hacerlos públicos) y subir materiales didácticos (PDFs, imágenes).
 - **Estudiantes:** Realizan exámenes con un cronómetro regresivo que guarda el progreso y reciben sus notas automáticamente o tras revisión manual.
3. **Inteligencia Artificial (El Factor Diferenciador):**
 - **Asistente Virtual:** Los estudiantes pueden seleccionar los materiales de su curso y hacerle preguntas a una IA que responde basándose únicamente en esos documentos (RAG básico).
 - **Generador de Flashcards:** Con un clic, la IA analiza los textos y crea tarjetas de estudio (pregunta/respuesta) para repasar conceptos clave.
 - **Feedback Automático:** Ayuda a los docentes generando sugerencias de corrección y retroalimentación constructiva para las respuestas de los alumnos.
4. **Administración y Auditoría:**

- Un panel para administradores que permite gestionar usuarios (activar/desactivar, cambiar roles), ver estadísticas globales del sitio y revisar un historial de seguridad (**AuditLog**) que registra quién hizo qué y desde qué IP.

Tecnologías Usadas (El "Stack")

El proyecto está construido sobre el stack **PERN** (Postgres, Express, React, Node):

- **Frontend:** React + Vite + TailwindCSS (para una interfaz rápida y moderna).
- **Backend:** Node.js + Express (lógica del servidor).
- **Base de Datos:** PostgreSQL + Prisma ORM (gestión eficiente de datos relacionales).
- **IA:** Integración con la API de OpenAI (GPT-3.5 Turbo).