

Manual Técnico

Introducción

Este manual técnico describe la arquitectura de la aplicación, el backend, la base de datos, y cómo interactúa con el frontend.

****Arquitectura del Proyecto:****

La aplicación sigue una arquitectura cliente-servidor utilizando:

- ****Frontend:**** Desarrollado en ReactJS (o Angular) para interactuar con los usuarios.
- ****Backend:**** Implementado en Node.js con Express para exponer las rutas de la API REST.
- ****Base de datos:**** MySQL para almacenar los usuarios, publicaciones, comentarios y cursos aprobados.

****Estructura de Rutas:****

- ****POST /auth/register****: Registrar un nuevo usuario.
- ****POST /auth/login****: Iniciar sesión y obtener un token JWT.
- ****POST /auth/forgot****: Enviar un enlace para recuperar la contraseña.
- ****POST /auth/reset****: Restablecer la contraseña del usuario.
- ****GET /posts****: Obtener todas las publicaciones, con filtros por curso o catedrático.
- ****POST /posts****: Crear una nueva publicación sobre un curso o catedrático.
- ****POST /comments/:postId****: Comentar una publicación.
- ****GET /users/me****: Ver el perfil de un usuario logueado.
- ****GET /users/by-registro/:ra****: Buscar a un usuario por su número de registro académico.

****Base de Datos:****

La base de datos usa MySQL y tiene las siguientes tablas:

- **users**: Contiene información de los usuarios (registro académico, nombre, email, contraseña).

- **posts**: Contiene las publicaciones creadas por los usuarios sobre cursos o catedráticos.

- **comments**: Contiene los comentarios sobre las publicaciones.

- **approved_courses**: Almacena los cursos aprobados por los usuarios.

Variables de Entorno:

El archivo `.env` debe contener las siguientes configuraciones:

- **DB_HOST**: Dirección del servidor de la base de datos.

- **DB_PORT**: Puerto del servidor MySQL (por defecto 3306).

- **DB_USER**: Usuario de la base de datos.

- **DB_PASSWORD**: Contraseña del usuario de la base de datos.

- **DB_NAME**: Nombre de la base de datos a utilizar.

- **PORT**: Puerto en el que corre la API (por defecto 4000).

Cómo ejecutar el proyecto:

1. Instalar dependencias: `npm install`

2. Configurar el archivo `.env` con las credenciales de la base de datos.

3. Iniciar el servidor: `npm run dev`

4. Acceder a la API desde `http://localhost:4000`.

Estructura de Archivos:

- **/src**: Contiene el código fuente del servidor (rutas, controladores, modelos).

- **/controllers**: Lógica de negocio para manejar las rutas.

- **/routes**: Definición de las rutas de la API.

- **/db.js**: Configuración de la conexión a la base de datos.

- **/utils**: Funciones auxiliares, como el hashing de contraseñas.

