

Historia de Usuario: Plataforma de Gestión de Cursos

Título:

Como administrador de una plataforma educativa, quiero poder gestionar usuarios, cursos y las inscripciones a los cursos, para que los estudiantes puedan inscribirse y acceder a sus clases correctamente.

Objetivo General:

Desarrollar un **backend completo** usando Node.js que permita manejar:

- Autenticación de usuarios con JWT
- Almacenamiento y relaciones entre tablas en una base de datos SQL mediante Sequelize
- Registro de logs o métricas en MongoDB
- Contenedorización del proyecto con Docker
- Automatización de tareas con Cron Jobs
- Pruebas unitarias con Jest
- Pruebas de endpoints con Postman (colección exportada)

Entidades principales (SQL con Sequelize):

1. Usuario

- a. id
- b. nombre
- c. correo
- d. contraseña (encriptada)
- e. rol (admin o estudiante)

2. Curso

- a. id
- b. título
- c. descripción
- d. fecha de inicio
- e. activo (boolean)

3. Inscripción

- a. id

- b. usuariold (FK)
- c. cursold (FK)
- d. fecha de inscripción

Relaciones:

- Un **usuario** puede estar inscrito en muchos **cursos**.
- Un **curso** puede tener muchos **usuarios** inscritos.
- Relación **N:M** entre Usuario y Curso a través de **Inscripción**.

Requerimientos técnicos:

Backend:

- Usar **Express.js** como framework base.
- Usar **Sequelize** como ORM para conectar con una base de datos relacional (PostgreSQL o MySQL).
- Implementar rutas CRUD para usuarios, cursos e inscripciones.
- Implementar **middleware de autenticación con JWT**.
 - Registro de usuarios
 - Login
 - Rutas protegidas según rol (admin o estudiante)

Docker:

- Crear un Dockerfile y docker-compose.yml para levantar:
 - El backend de Node.js
 - La base de datos SQL
 - MongoDB (ver siguiente punto)

MongoDB:

- Crear una colección logs en MongoDB donde se guarden:
 - Cada vez que un usuario se inscribe en un curso
 - Cada vez que un curso es creado o editado
- Estos logs deben guardarse en paralelo al flujo principal del backend (puede usarse un helper o middleware).

Cron Job:

- Crear un cron job que:
 - Se ejecute una vez al día
 - Verifique los cursos con fecha de inicio anterior a la actual y los desactive (activo = false)
 - Genere un log de los cursos desactivados

Tests:

- Escribir al menos **5 pruebas unitarias con Jest** (por ejemplo, lógica de inscripciones, login, etc.)
- Exportar una colección de **Postman** con pruebas de endpoints:
 - Crear usuario
 - Login
 - Crear curso (requiere token de admin)
 - Inscribirse en curso
 - Ver cursos inscritos