

Gestor de Notas - Documentación

Descripción del Proyecto

El Gestor de Notas es una aplicación web que permite a los usuarios registrarse, iniciar sesión y gestionar notas personales. Incluye un sistema CRUD completo (Crear, Leer, Actualizar y Eliminar notas), autenticación segura mediante JWT, y roles de usuario como administrador y usuario.

La aplicación se compone de un backend (Node.js y Express) y un frontend (React y TailwindCSS).

Tecnologías Utilizadas

Backend:

- Node.js: Entorno de ejecución de JavaScript.
- Express: Framework para crear el servidor web.
- MySQL: Base de datos para almacenar usuarios y notas.
- Bcrypt: Para cifrar contraseñas.
- JWT (JSON Web Tokens): Autenticación y manejo de sesiones.
- Dotenv: Gestión de variables de entorno.

Frontend:

- React: Biblioteca para construir la interfaz de usuario.
- React Router: Manejo de navegación entre páginas.
- Axios: Cliente HTTP para conectar con el backend.
- TailwindCSS: Framework para estilos CSS.

Estructura del Proyecto

...

gestor-notas/

```
|
├── backend/      # Código del servidor
│   ├── controllers/  # Controladores para manejar lógica de negocio
│   ├── models/      # Consultas SQL a la base de datos
│   ├── routes/      # Rutas del servidor (endpoints)
│   ├── middleware/   # Middleware para autenticación
│   ├── config/      # Configuraciones globales
│   ├── index.js     # Archivo principal del servidor
│   └── .env         # Variables de entorno (puerto, base de datos, etc.)
├── frontend/      # Código del cliente
│   └── public/     # Archivos estáticos
```

```

├── src/          # Código React principal
│   ├── components/ # Componentes reutilizables
│   ├── pages/     # Páginas (Home, Login, Dashboard)
│   ├── App.js     # Configuración principal
│   └── index.js   # Punto de entrada de React
├── package.json  # Dependencias del frontend
├── tailwind.config.js # Configuración de Tailwind
└── README.md     # Documentación
...

```

Backend: Lógica Principal

1. Autenticación de Usuarios

La autenticación se maneja con JWT para generar tokens únicos que permiten identificar al usuario.

- **Registrar Usuario**: Crea un usuario y cifra la contraseña usando bcrypt. Si es el primer usuario registrado, se asigna automáticamente el rol de admin.

```

```javascript
static async createUser(name, email, password) {
 const hashedPassword = await bcrypt.hash(password, 10);
 const isFirstUser = await db.query('SELECT COUNT(*) FROM user');
 const role = isFirstUser[0][0]['COUNT(*)'] === 0 ? 'admin' : 'user';
 const query = 'INSERT INTO user (name, email, password, role) VALUES (?, ?, ?, ?)';
 return db.query(query, [name, email, hashedPassword, role]);
}
...

```

### 2. Gestión de Notas (CRUD)

Permite realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) notas asociadas a un usuario autenticado.

## Instalación y Configuración

### 1. Clonar el Proyecto

```

```bash
git clone https://github.com/tuusuario/gestor-de-notas.git
cd gestor-de-notas
...

```

2. Backend

1. Instala dependencias:

```

```bash
cd backend

```

```
npm install
```

```
npm install
```

```
npm install express
```

```
npm install mysql2
```

```
npm install dotenv
```

```
npm install bcrypt
```

```
npm install jsonwebtoken
```

```
npm install cors
```

```
npm install --save-dev nodemon
```

```
...
```

2. Configura el archivo `.env` con tus credenciales:

```
``plaintext
```

```
PORT=2554
```

```
DB_HOST=localhost
```

```
DB_USER=root
```

```
DB_PASSWORD=12345
```

```
DB_NAME=notas_project
```

```
JWT_SECRET=tuClaveSecreta
```

```
...
```

3. Ejecuta el servidor:

```
``bash
```

```
npm run dev
```

```
...
```

### 3. Frontend

1. Instala dependencias:

```
``bash
```

```
cd frontend
```

```
npm install
```

```
npm install -D tailwindcss postcss autoprefixer
```

```
npx tailwindcss init -p
```

```
npm install react react-dom
```

```
npm install react-router-dom
```

```
npm install axios
```

```
'''
```

2. Ejecuta la aplicación:

```
```bash
```

```
npm start
```

```
'''
```