**Informe Técnico del Proyecto Full-Stack: Blog App**

**1. Introducción**

El presente informe técnico documenta el desarrollo de una aplicación web con arquitectura full-stack, la cual permite a los usuarios registrarse, iniciar sesión y gestionar publicaciones tipo blog. El proyecto fue desarrollado utilizando React para el front-end, Node.js y Express para el back-end, y MongoDB como base de datos.

# 2. Estructura de la Aplicación Web

La arquitectura está dividida en dos partes principales:  
- Front-End (React): interfaz de usuario, navegación y formularios.  
- Back-End (Node.js/Express): API RESTful, lógica de negocio y conexión a MongoDB.  
- Base de Datos (MongoDB): almacenamiento de usuarios y publicaciones.

# 3. Configuración del Servidor y Conexión a MongoDB

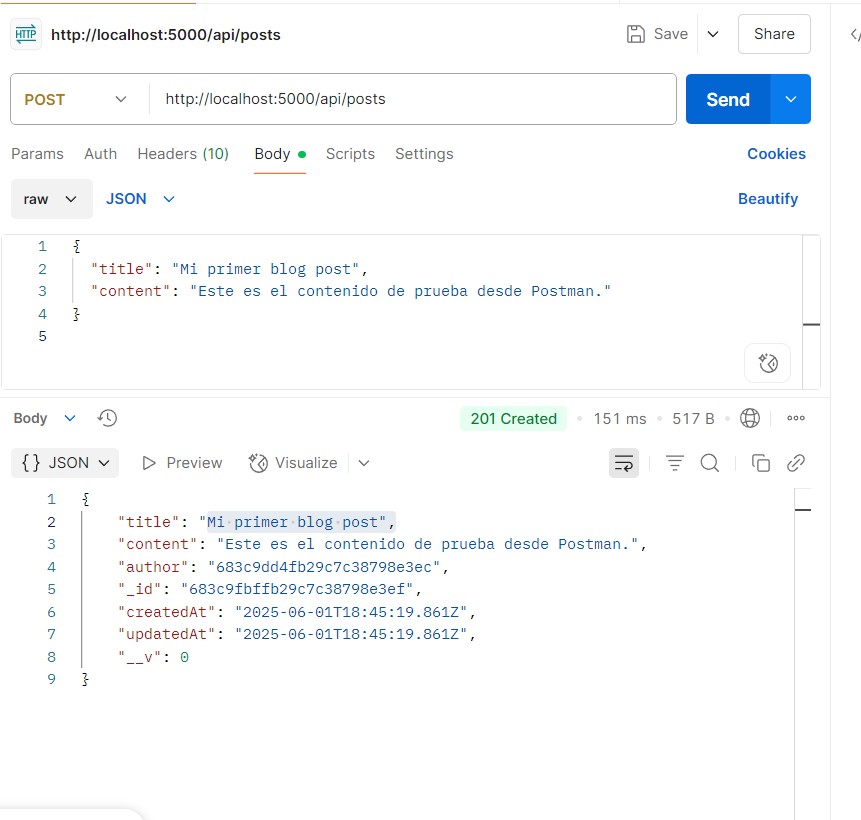
El servidor fue configurado utilizando Express en Node.js. Se definieron rutas para autenticar usuarios y gestionar publicaciones.  
La conexión a la base de datos MongoDB se realizó utilizando Mongoose, definiendo esquemas para usuarios y publicaciones.  
La autenticación se realizó mediante JSON Web Tokens (JWT).

# 4. Implementación del Front-End con React y Redux

El front-end se desarrolló con React y se organizaron los componentes por páginas: registro, inicio de sesión, creación de publicaciones, y visualización de posts.  
Para el manejo del estado de autenticación y datos del blog se implementó Redux, lo cual permite gestionar los tokens de sesión y acceder a ellos globalmente desde cualquier componente.

# 5. Flujo de Trabajo y Herramientas Utilizadas

El proyecto se inició creando el entorno con Node.js y Create React App. Se utilizó Visual Studio Code como editor de código y GitHub para control de versiones.  
Las herramientas y tecnologías utilizadas fueron:  
- React  
- Node.js  
- Express  
- MongoDB  
- Mongoose  
- Redux  
- Postman  
- Git y GitHub



Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.



Texto

El contenido generado por IA puede ser incorrecto.

# 6. Conclusión

El proyecto fue completado exitosamente. Se implementó el registro y login de usuarios, autenticación JWT, almacenamiento de publicaciones, y una interfaz clara.  
El despliegue local fue probado correctamente, y el código se encuentra disponible en GitHub para su revisión o futura mejora.

# 7. Referencias

American Psychological Association. (2020). \*Publication manual of the American Psychological Association\* (7.ª ed.). APA.  
Mozilla Developer Network. (2024). \*Express.js documentation\*. https://developer.mozilla.org/  
React Documentation. (2024). \*React – A JavaScript library for building user interfaces\*. https://reactjs.org  
MongoDB, Inc. (2024). \*Mongoose documentation\*. https://mongoosejs.com  
Redux.js. (2024). \*Redux official documentation\*. https://redux.js.org  
GitHub. (2024). \*Repositorio del proyecto Blog App\*. https://github.com/luishernandez27/blog-app