

Nombre: Quenguan Luna Ever Yesid

Correo: Hevershtp@gmail.com

Celular: 3178920038 (Whatsapp) – 3154495936

Informe del Proyecto: Aplicación de Notas "ToDo"

1. Introducción

Objetivo:

Este proyecto tiene como objetivo desarrollar una aplicación de gestión de notas "ToDo" utilizando **Laravel** y **Vue.js**, siguiendo las mejores prácticas de desarrollo web. La aplicación permite la creación, edición, eliminación y listado de notas, con funcionalidades adicionales como autenticación de usuarios y manejo de etiquetas.

2. Tecnologías Utilizadas

- **Backend:** Laravel 8
- **Frontend:** Vue.js 2.7
- **Motor de base de datos:** MySQL 8
- **Autenticación:** Laravel Sanctum para tokens y autenticación de usuario
- **Framework CSS:** TailwindCSS
- **Sistema de Versionamiento:** https://github.com/Hever-Quenguan/Proyecto_Notas-Laravel-.git

3. Estructura del Proyecto

- **Carpetas principales:**
 - app/: Lógica del backend en Laravel.
 - resources/js/: Código de frontend utilizando Vue.js.
 - routes/: Definición de rutas del backend (API y web).
 - database/: Migraciones y seeding para la base de datos.
 - public/: Recursos estáticos del frontend (imágenes, CSS, etc.).

Organización de archivos

El proyecto se ha organizado siguiendo las buenas prácticas de Laravel, separando de forma clara las capas de lógica, rutas y vistas. En el frontend, las rutas están gestionadas con **Vue Router** y el estado con **Vuex**.

4. Backend

Lógica del Módulo de Notas

La aplicación permite realizar las operaciones básicas (CRUD) sobre las notas. Estas operaciones son manejadas a través de un **API RESTful**, y la lógica se encapsula en controladores y servicios reutilizables.

API REST

La API REST fue implementada usando los siguientes métodos:

- **GET /api/notes:** Para obtener el listado de notas.
- **POST /api/notes:** Para crear una nueva nota.
- **PUT /api/notes/{id}:** Para actualizar una nota existente.
- **DELETE /api/notes/{id}:** Para eliminar una nota.

Las operaciones están protegidas mediante autenticación de usuarios utilizando **Laravel Sanctum**.

Autenticación

- Se implementó la autenticación de usuarios mediante tokens utilizando **Laravel Passport**
- Los usuarios pueden registrarse e iniciar sesión para gestionar sus notas.

Validación de Datos

- Se utilizan **Request Classes** para validar los datos tanto en la creación como en la edición de notas.
- La validación incluye campos requeridos como título, descripción, usuario, entre otros.

Migraciones

- Las tablas de la base de datos fueron generadas mediante migraciones de Laravel, asegurando una estructura clara y normalizada.

Tablas principales:

Comandos Artisan

5. Frontend

Pantallas Implementadas

- **Pantalla de Inicio de Sesión y Registro:** Los usuarios pueden iniciar sesión o registrarse utilizando un formulario sencillo. El estado del usuario se maneja con **Vuex** para mantener la sesión activa.
- **Pantalla Principal de Notas:**
 - Listado de todas las notas del usuario autenticado, ordenadas por **fecha de creación** o **fecha de vencimiento**.
 - Cada nota muestra el título, descripción, etiquetas y otros detalles relevantes.
- **Formulario de Creación y Edición de Notas:** Se ha implementado un formulario donde los usuarios pueden agregar nuevas notas o editar las existentes. Los campos como título, descripción, usuario y etiquetas son obligatorios.

Peticiones AJAX

Las peticiones AJAX para interactuar con la API se realizaron utilizando **Axios**. Esto asegura una comunicación eficiente entre el frontend y el backend, permitiendo la creación, edición y eliminación de notas sin recargar la página.

Vue Router y Vuex

- **Vue Router** se utilizó para manejar las diferentes rutas de la aplicación, como el listado de notas, el formulario de edición y la página de inicio de sesión.
- **Vuex** gestiona el estado global de la aplicación, incluyendo la autenticación del usuario.

6. Validación de Datos en Frontend

Se realizaron validaciones de los formularios en el frontend para evitar el envío de datos incompletos o incorrectos, usando las capacidades nativas de Vue.js.

7. Conclusión

El proyecto cumple con los requerimientos y condiciones establecidas, implementando las mejores prácticas de desarrollo tanto en el backend como en el frontend. Se ofrece una experiencia de usuario intuitiva y funcional, además de garantizar la seguridad y validación adecuada de los datos.

Posibles Mejoras Futuras

- Implementación de notificaciones en tiempo real.
- Mejora de la interfaz de usuario con más personalización.
- Implementación de pruebas unitarias para garantizar la robustez del sistema.

8. Enlaces del Proyecto

- **Repositorio del código fuente:** https://github.com/Hever-Quenguan/Proyecto_Notas-Laravel-.git