

1. Introducción

Este proyecto es una aplicación web que permite la gestión de blogs, usuarios y productos, integrando funcionalidades como autenticación de usuarios, manejo de una base de datos en línea, y procesamiento de pagos a través de PayPal. La aplicación está desarrollada utilizando Node.js con Express, MongoDB como base de datos, y EJS como motor de plantillas.

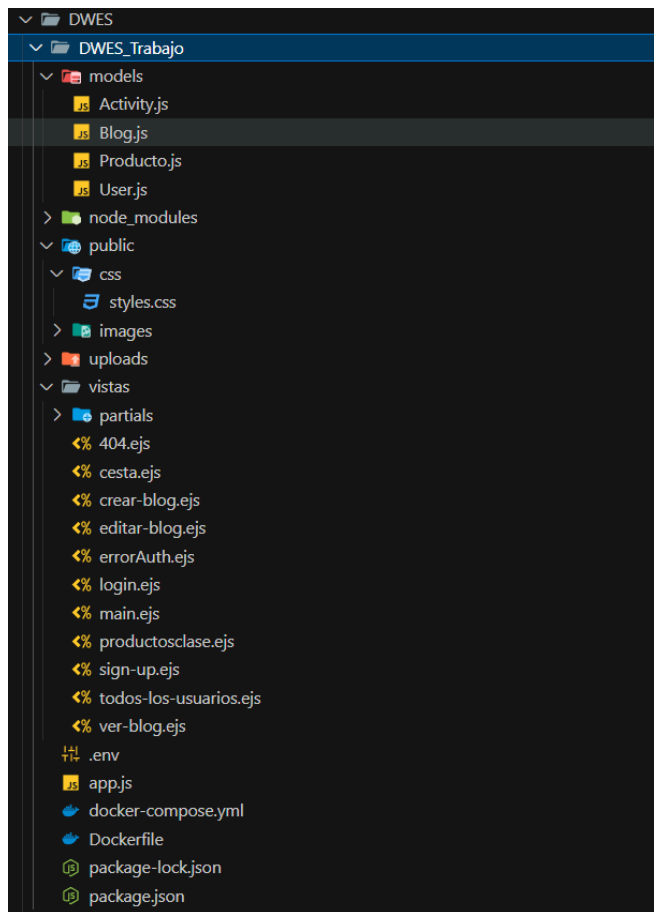
2. Funcionamiento General

La aplicación permite a los usuarios registrarse, iniciar sesión, y acceder a rutas protegidas donde pueden gestionar blogs y productos. Los usuarios pueden crear, leer, actualizar y eliminar blogs (CRUD), así como agregar productos a una cesta de compras y realizar pagos a través de PayPal. La autenticación se maneja con Passport.js, y las sesiones se gestionan con express-session.

3. Estructura del proyecto

El proyecto está organizado en los siguientes directorios y archivos:

- app.js: Archivo principal de la aplicación donde se configura el servidor, las rutas, y los middlewares.
- models/: Contiene los modelos de Mongoose para User, Blog, Producto, y Activity.
- vistas/: Contiene las plantillas EJS para las vistas de la aplicación.
- partials/: Incluye componentes reutilizables como header.ejs y footer.ejs.
- main.ejs: Página principal que muestra los últimos blogs.
- login.ejs: Formulario de inicio de sesión.
- sign-up.ejs: Formulario de registro de usuarios.
- todos-los-usuarios.ejs: Lista de todos los usuarios registrados.
- productosclase.ejs: Lista de productos disponibles.
- cesta.ejs: Vista de la cesta de compras.
- crear-blog.ejs: Formulario para crear un nuevo blog.
- editar-blog.ejs: Formulario para editar un blog existente.
- ver-blog.ejs: Vista detallada de un blog.
- errorAuth.ejs: Página de error de autenticación.
- 404.ejs: Página de error 404.
- public/: Contiene archivos estáticos como CSS.
- styles.css: Estilos CSS para la aplicación.
- uploads/: Directorio donde se almacenan las imágenes subidas por los usuarios.



4. Clases y Modelos

User: Modelo de Mongoose para los usuarios. Incluye campos como nombre, apellido, email, password, edad, nacionalidad, sexo, birthdate, y imagePath.

Blog: Modelo de Mongoose para los blogs. Incluye campos como title, snippet, body, fecha, imagePath, y createdAt.

Producto: Modelo de Mongoose para los productos. Incluye campos como nombre, descripcion, precio, y imageUrl.

Activity: Modelo de Mongoose para actividades (no implementado en el código proporcionado).

5. Flujo de Funcionamiento

- Carga de la página: Al cargar la página principal (/), se muestran los últimos blogs ordenados por fecha de creación.
- Autenticación: Los usuarios pueden registrarse (/sign-up) e iniciar sesión (/login). Una vez autenticados, pueden acceder a rutas protegidas como /todos-los-usuarios y /blogs/crear.
- Gestión de Blogs: Los usuarios autenticados pueden crear, editar, y eliminar blogs. Los blogs se almacenan en la base de datos y se muestran en la página principal.
- Gestión de Productos: Los usuarios pueden ver una lista de productos (/productos), agregarlos a la cesta, y realizar pagos a través de PayPal.
- Pagos con PayPal: Los usuarios pueden pagar los productos en la cesta utilizando PayPal. El pago se procesa a través de la API de PayPal, y la cesta se limpia después de un pago exitoso.

6. Funcionalidad Adicional

- Subida de Imágenes: Los usuarios pueden subir imágenes al crear o editar blogs y al registrarse. Las imágenes se almacenan en el servidor y se muestran en las vistas correspondientes.
- Manejo de Sesiones: Las sesiones de los usuarios se mantienen durante 20 minutos de inactividad.
- Variables de Entorno: Se utiliza un archivo .env para almacenar variables sensibles como la URI de MongoDB y las credenciales de PayPal.