**ESTRUCTURA DE UN PROYECTO WEB**

Integrantes

Hemer Santiago Perez Nieves, Hugo Andres Forero, Miguel Angel Urrea Camacho, Miguel Angel Pinilla Baez y Yesid Alejandro Varela Alfa

Institución Educativa, Universidad Manuela Beltran

Docente, Diana Toquica

Asignatura, Ingeniería Web

Facultad, Ingeniería de Software

Bogota D.c

4 de Febrero, 2025

**Definir la arquitectura:**

**Framework:**

**Node.js**: Node.js es una plataforma de ejecución de JavaScript del lado del servidor. Gracias a su modelo de I/O no bloqueante y basado en eventos, permite construir aplicaciones rápidas y escalables que gestionan múltiples solicitudes simultáneas de manera eficiente. En tu proyecto de página web de libros, Node.js se encarga de la lógica de backend, gestionando las interacciones con la base de datos, la autenticación de usuarios, las peticiones HTTP, y proporcionando el rendimiento necesario para soportar un alto volumen de tráfico.

**Herramientas de testeo:**

**Postman**: Postman es fundamental para probar las APIs antes de integrarlas en el frontend. Te permite enviar solicitudes HTTP a tu backend y recibir las respuestas para verificar que los endpoints estén funcionando como se espera. Puedes probar rutas de libros, autenticación de usuarios, y validación de datos de manera controlada y precisa.

**RestClient**: Similar a Postman, pero se integra directamente en Visual Studio Code, RestClient permite realizar pruebas de APIs sin salir de tu entorno de desarrollo. Esto te permite probar rápidamente tus endpoints durante el proceso de desarrollo sin necesidad de cambiar de aplicación o entorno.

**Base de datos:**

**PostgreSQL**: PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) que proporciona una gran robustez, flexibilidad y escalabilidad. Es ideal para manejar datos complejos, como los libros, autores y registros de usuarios. Ofrece potentes capacidades de consulta y permite trabajar con tipos de datos como JSON o arrays, lo que es útil para almacenar metadatos o características adicionales de los libros. También se destaca por ser una base de datos altamente compatible con transacciones y por su alta disponibilidad, ideal para aplicaciones que manejan grandes volúmenes de información.

**Lenguaje de programación:**

**JavaScript**: JavaScript es un lenguaje que se utiliza tanto en el frontend como en el backend (mediante Node.js). Esta versatilidad simplifica la estructura del proyecto, permitiendo un desarrollo más ágil y menos propenso a errores. En el frontend, JavaScript permite interactuar con los usuarios, mostrando información sobre libros de forma dinámica y atractiva. En el backend, con Node.js, maneja la lógica del servidor, gestionando bases de datos, autenticación de usuarios y las rutas del API.

**Implementación y autenticación:**

**JWT (JSON Web Token)**: JWT es ideal para la autenticación sin estado, lo que significa que no necesitas almacenar información de sesión en el servidor. Cada vez que el usuario inicia sesión, se le entrega un token JWT que contiene información de autenticación y se incluye en cada solicitud posterior. Esto mejora la escalabilidad y la seguridad de tu aplicación web de libros, ya que no se requiere gestionar sesiones en el servidor y la validación del token se realiza de manera rápida y segura.

**bcrypt**: bcrypt es una librería utilizada para cifrar las contraseñas de los usuarios antes de almacenarlas en la base de datos. Este proceso convierte las contraseñas en un formato irreversible (hash), lo que hace que incluso si los datos de la base de datos se ven comprometidos, las contraseñas no puedan ser recuperadas ni utilizadas maliciosamente. Este enfoque mejora la seguridad de la página web de libros.

**Express Validator**: Express Validator se encarga de validar y sanitizar los datos que el servidor recibe. Por ejemplo, verifica que los correos electrónicos sean válidos, que las contraseñas cumplan con los requisitos de longitud y complejidad, o que los datos enviados por los usuarios estén libres de caracteres especiales o maliciosos que podrían comprometer la seguridad de la aplicación.

**Dotenv**: Dotenv es útil para cargar y manejar variables de entorno como claves de API, configuraciones de base de datos, o credenciales en diferentes entornos (desarrollo, producción). Utilizando un archivo .env, puedes mantener estas configuraciones fuera del código fuente y asegurar que no se expongan, mejorando así la seguridad.

**Optimización de consultas:**

**Prisma**: Prisma es un ORM (Object-Relational Mapping) moderno que facilita la interacción con bases de datos relacionales. A través de su cliente, puedes realizar operaciones CRUD de manera sencilla y eficiente sin escribir SQL manualmente. Prisma también te permite manejar relaciones entre tablas (por ejemplo, entre libros y autores) de manera más intuitiva, mejorando la productividad y la legibilidad del código. Al ser un ORM altamente optimizado, Prisma ayuda a evitar consultas innecesarias o ineficientes y mejora el rendimiento en las operaciones de lectura y escritura en la base de datos.