ESTRUCTURA DE UN PROYECTO WEB

Integrantes

Hemer Santiago Pérez Nieves, Hugo Andrés Forero, Miguel Ángel Urrea Camacho,

Miguel Ángel Pinilla Báez y Yesid Alejandro Varela Alfaro

Institución Educativa, Universidad Manuela Beltrán

Docente, Diana Toquica

Asignatura, Ingeniería Web

Facultad, Ingeniería de Software

Bogotá D.C.

4 de febrero, 2025

1. Node.js

Node.js es el motor central del backend de la aplicación, ejecutando el código JavaScript en el servidor. Aprovecha su naturaleza no bloqueante y basada en eventos para manejar múltiples solicitudes de usuarios de forma concurrente y eficiente, lo que lo convierte en una opción ideal para una aplicación web con alto volumen de tráfico, como una plataforma de libros.

Conexión con otras herramientas:

Express.js: Node.js se apoya en Express.js como un framework minimalista y flexible para la creación de APIs RESTful. Express facilita la configuración de rutas, controladores y middlewares para manejar las solicitudes HTTP, lo que permite interactuar con el frontend.

Prisma: A través de Prisma, Node.js se conecta a la base de datos PostgreSQL. Prisma actúa como un ORM para realizar operaciones CRUD de manera sencilla y eficiente sin necesidad de escribir SQL directamente.

2. Base de Datos: PostgreSQL

PostgreSQL es una base de datos relacional que gestiona la información clave de la plataforma, como los libros, autores y usuarios. La estructura de tablas y relaciones entre datos (como los autores de los libros) se maneja en esta base de datos, y PostgreSQL es especialmente adecuada para proyectos que requieren escalabilidad, consistencia y transacciones complejas.

Conexión con otras herramientas:

Prisma ORM: Prisma se conecta directamente a PostgreSQL, proporcionándole un cliente que abstrae las consultas SQL, haciendo que la interacción con la base de datos sea más fácil y más segura. Prisma facilita la creación de relaciones entre las tablas, como entre libros y autores, permitiendo una navegación fluida entre los datos.

3. Herramientas de Testeo: Postman y RestClient

Postman y RestClient son herramientas esenciales para probar y depurar las API desarrolladas en el backend. Permiten enviar solicitudes HTTP al servidor y verificar que los endpoints y las funcionalidades de la aplicación están trabajando correctamente antes de integrar el frontend.

Conexión con otras herramientas:

Postman: Se utiliza para probar las rutas de la API que interactúan con la base de datos y el servidor de Node.js. Por ejemplo, se pueden enviar solicitudes para agregar un libro o consultar todos los libros en la base de datos, asegurando que los endpoints proporcionados por Express.js estén funcionando correctamente.

RestClient: Esta herramienta se integra directamente en el entorno de desarrollo (Visual Studio Code), lo que permite a los desarrolladores realizar pruebas rápidas de las API sin necesidad de cambiar de aplicación. Esto mejora la productividad durante la fase de desarrollo, ya que no es necesario alternar entre diferentes interfaces.

4. Autenticación y Seguridad: JWT, bcrypt, y Express Validator

La seguridad es fundamental en cualquier aplicación web, y en este caso, las herramientas de autenticación y validación juegan un papel importante en la protección de los datos del usuario y el acceso a la aplicación.

Conexión con otras herramientas:

JWT (JSON Web Token): JWT se utiliza para la autenticación sin estado. Cuando un usuario inicia sesión, el servidor genera un token JWT, que incluye información como el ID del usuario. Este token es enviado al cliente y se incluye en las solicitudes HTTP subsecuentes para verificar que el usuario tiene acceso a las rutas protegidas. JWT ayuda a escalar la aplicación sin necesidad de almacenar sesiones en el servidor.

bcrypt: Cuando un usuario crea una cuenta o cambia su contraseña, bcrypt se encarga de cifrar la contraseña antes de almacenarla en la base de datos. Este proceso asegura que las contraseñas nunca se guardan en texto claro, protegiendo así la seguridad de los datos del usuario en caso de un ataque a la base de datos.

Express Validator: Express Validator se usa para validar y sanitizar los datos de entrada del usuario antes de que sean procesados por el servidor. Por ejemplo, se pueden validar correos electrónicos y contraseñas para asegurarse de que cumplen con los requisitos establecidos (longitud mínima, complejidad, etc.). Esto ayuda a evitar ataques como la inyección de SQL y asegura la calidad y consistencia de los datos.

5. Gestión de Variables de Entorno: Dotenv

Dotenv se utiliza para manejar variables de entorno de manera segura y eficiente. En el archivo `.env`, se almacenan datos sensibles como las credenciales de la base de datos, claves de API, y configuraciones específicas para diferentes entornos (desarrollo, producción).

Conexión con otras herramientas:

Node.js y Express: Dotenv carga las variables de entorno al iniciar el servidor de Node.js, lo que permite acceder a ellas dentro del código sin necesidad de incluirlas explícitamente en el código fuente. Por ejemplo, las claves de API o las credenciales de la base de datos se mantienen seguras y no se exponen en el repositorio de código.

6. Optimización de Consultas: Prisma ORM

Prisma es un ORM moderno que se utiliza para interactuar con PostgreSQL de manera más eficiente y legible. Prisma proporciona un cliente de base de datos que abstrae la escritura de SQL manual, facilitando las operaciones CRUD y mejorando la productividad.

Conexión con otras herramientas:

Node.js: Prisma se integra con el backend de Node.js para realizar las consultas a la base de datos. Las relaciones entre los libros y los autores, así como las consultas complejas, se gestionan mediante Prisma, lo que evita la necesidad de escribir consultas SQL manualmente.

PostgreSQL: Prisma permite realizar consultas a PostgreSQL de manera optimizada, ayudando a evitar consultas ineficientes y mejorando el rendimiento general de la aplicación, especialmente en operaciones de lectura y escritura de grandes volúmenes de datos.

Bases de Datos