Thiago Henrique Benedito Alves – ADS Turma B

Parte 1: Perguntas Teoricas

Essas são ótimas perguntas para entender melhor as tecnologias e práticas em um sistema backend.

1. O que é Node.js e por que ele é uma boa escolha para o desenvolvimento do backend de uma aplicação web?

Node.js é um ambiente de execução JavaScript que permite executar código JavaScript no servidor. Construído sobre o motor V8 do Chrome, ele é conhecido por sua eficiência e desempenho, especialmente em sistemas de entrada/saída (I/O) de alta demanda, devido ao seu modelo assíncrono e orientado a eventos.

Por que escolher Node.js para o backend?

- Desempenho: Node.js lida bem com grandes volumes de requisições simultâneas devido ao seu modelo assíncrono e não-bloqueante.

- JavaScript full-stack: Como usa JavaScript, é possível utilizar a mesma linguagem tanto no frontend quanto no backend, facilitando o desenvolvimento e a manutenção.

- Ecosistema robusto: A comunidade Node.js oferece muitos pacotes e módulos, acessíveis pelo npm (Node Package Manager), o que acelera o desenvolvimento.

Essas características fazem do Node.js uma escolha robusta e eficiente para o backend de aplicações web, especialmente em sistemas de microserviços e APIs RESTful.

2. O que é o MongoDB e como ele difere de um banco de dados relacional? Quais são as vantagens de usar MongoDB para o gerenciamento de dados da biblioteca?

MongoDB é um banco de dados NoSQL orientado a documentos que armazena dados em documentos JSON ou BSON, ao invés das tabelas e linhas de bancos de dados relacionais (como MySQL ou PostgreSQL).

Diferenças principais entre MongoDB e bancos de dados relacionais:

- Modelo de dados flexível: MongoDB permite armazenar dados em documentos que podem ter diferentes estruturas, o que facilita a evolução do esquema de dados sem a necessidade de migrações complexas.

- Escalabilidade horizontal: MongoDB é projetado para escalar facilmente em clusters distribuídos.

- Desempenho em consultas complexas: MongoDB oferece uma velocidade de leitura e escrita eficiente, especialmente para dados que não exigem joins complexos entre tabelas.

Vantagens para o gerenciamento de dados de biblioteca:

- Modelo flexível: Como o MongoDB permite mudanças de esquema de forma dinâmica, ele é ideal para aplicações que lidam com dados variados, como livros, autores e usuários.

- Alta disponibilidade e escalabilidade: O MongoDB pode crescer com facilidade para atender a demandas maiores, útil para bibliotecas que expandem seu acervo e base de usuários.

- Desempenho de leitura e escrita: MongoDB é eficiente para consultas simples e operações de leitura e escrita, ideais para registros de livros e usuários em bibliotecas.

3. Descreva o propósito do Mongoose no desenvolvimento de uma aplicação com MongoDB. Como ele facilita a interação com o banco de dados?

Mongoose é uma biblioteca Node.js que fornece uma camada de abstração para interagir com o MongoDB. Ele permite a criação de esquemas de dados para documentos no MongoDB e simplifica a comunicação entre a aplicação e o banco de dados.

Facilidades oferecidas pelo Mongoose:

- Esquemas e modelos: Mongoose permite definir modelos de dados que ajudam a manter a consistência e integridade dos dados, mesmo em um banco NoSQL.

- Validação de dados: É possível configurar validações e constraints para garantir que apenas dados válidos sejam salvos.

- Métodos e middlewares: Mongoose permite definir métodos personalizados para modelos, facilitando a organização da lógica de negócios. Além disso, é possível adicionar middlewares para manipular dados antes de salvar, atualizar ou excluir.

- Query simplificada: Mongoose abstrai operações de consulta complexas, tornando o código mais limpo e fácil de gerenciar.

Com essas funcionalidades, o Mongoose torna o desenvolvimento com MongoDB mais seguro e estruturado.

4. Quais são as principais funcionalidades que uma API RESTful deve oferecer? Por que essas funcionalidades são importantes para o sistema de gerenciamento de biblioteca?

Uma API RESTful é um estilo de arquitetura que facilita a criação de sistemas de fácil manutenção e escalabilidade. Para um sistema de gerenciamento de biblioteca, uma API RESTful deve oferecer:

- Operações CRUD (Create, Read, Update, Delete): Permitir a criação, leitura, atualização e exclusão de registros, como livros, autores e usuários.

- Endpoints bem definidos e semânticos: Endpoints que refletem a estrutura de dados (ex: `/api/books`, `/api/authors`) facilitam a organização e uso da API.

- Autenticação e autorização: Garantir que apenas usuários autorizados possam acessar ou modificar certos dados, especialmente informações sensíveis.

- Paginação e filtragem: Para consultas eficientes, principalmente em acervos grandes de livros, é essencial implementar paginação e permitir que usuários filtrem dados.

- Mensagens de erro adequadas: Mensagens de erro claras ajudam na depuração e fornecem uma experiência melhor aos desenvolvedores que utilizam a API.

Essas funcionalidades são importantes para garantir que o sistema de biblioteca seja eficiente, seguro e fácil de escalar.

5. Quais práticas você deve seguir para garantir a segurança de uma aplicação web que manipula dados sensíveis, como informações sobre livros e usuários?

Para proteger dados sensíveis em uma aplicação web, as seguintes práticas de segurança são essenciais:

- Autenticação e autorização: Utilizar autenticação JWT ou OAuth para assegurar que apenas usuários autorizados possam acessar informações confidenciais.

- Criptografia de senhas: Usar hashing seguro (ex: bcrypt) para armazenar senhas, evitando que sejam armazenadas em texto plano.

- Validação e sanitização de dados: Validar e sanitizar todos os dados de entrada para evitar injeção de SQL e ataques XSS.

- HTTPS: Utilizar HTTPS para encriptar dados transmitidos entre o cliente e o servidor, especialmente em endpoints que lidam com dados sensíveis.

- Rate limiting: Implementar limites de requisição por IP para evitar ataques de força bruta e reduzir o impacto de acessos não autorizados.

- Logs e monitoramento: Registrar atividades no sistema e monitorar possíveis padrões de ataque para detectar rapidamente atividades suspeitas.