Nome: Leonardo Rodrigues – ADS – Turma B – 2º Semestre

Questões Teóricas

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

R: Node.js é um ambiente de execução JavaScript no lado do servidor, baseado no motor V8 do Google Chrome. Ele permite que você escreva aplicações web em JavaScript no servidor, utilizando um único idioma tanto no cliente quanto no servidor.

• Explique 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?

R: MongoDB é um banco de dados NoSQL orientado a documentos. Ele armazena dados em formato BSON (uma forma binária de JSON), em vez de tabelas e linhas como um banco de dados relacional.

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

R: Mongoose é uma biblioteca para Node.js que fornece uma solução baseada em esquemas para interagir com o MongoDB. Ele facilita a definição de modelos de dados, validação, criação e manipulação de documentos.

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

R: Funcionalidades principais:

GET: Para listar ou buscar livros na biblioteca.

POST: Para adicionar novos livros à biblioteca.

PUT/PATCH: Para atualizar informações de um livro existente.

DELETE: Para remover livros da biblioteca.

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

R: Autenticação: Só permite o acesso a quem tem permissão. Usa senhas ou sistemas como login com código (ex: login com Google).

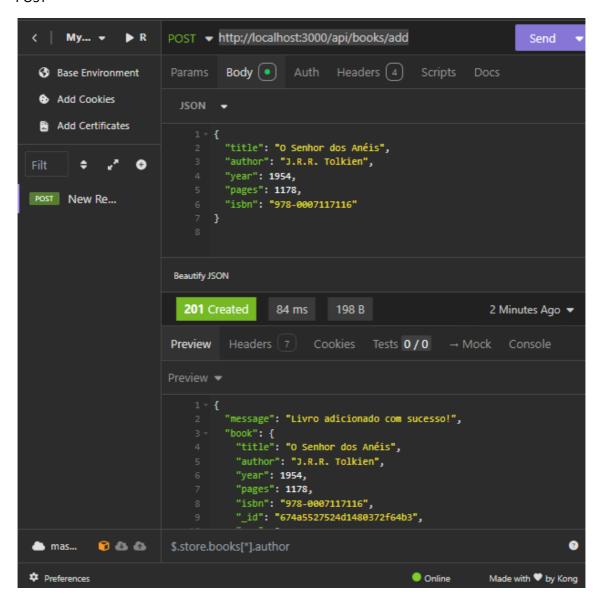
Proteção de Senhas: Não armazena senhas em texto simples. Usa técnicas que embaralham a senha (ex: bcrypt) para que, mesmo se alguém roubar os dados, não consiga ler a senha.

Evitar Ataques: Protejer a aplicação contra ataques comuns, como spam e invasões, usando sistemas de proteção como anti-cross site.

Verificar Entradas: Garantir que os dados inseridos pelos usuários sejam válidos e seguros para evitar problemas, como roubo de dados.

TESTES:

POST



DELETE

