

1. Defina o conceito de Vue.js e cite pelo menos três de suas principais características.

Vue.js é um framework JavaScript para criar interfaces interativas.

Características: reatividade, componentização e fácil integração.

2. Explique a importância de usar Node.js em aplicações web. Como ele difere de outras tecnologias de back-end como PHP?

Node.js é importante por ser leve, rápido e eficiente, ideal para aplicações em tempo real como chats e APIs. Ele usa JavaScript no servidor e possui I/O não bloqueante, o que melhora a performance.

Diferença principal: Node.js é assíncrono e orientado a eventos, enquanto o PHP é síncrono e processa uma requisição por vez, o que pode tornar o desempenho inferior em aplicações com muitas conexões simultâneas.

3. O que são APIs RESTful? Cite suas principais vantagens no desenvolvimento de sistemas modernos.

APIs RESTful são interfaces que seguem os princípios REST, permitindo comunicação entre sistemas usando o protocolo HTTP.

Vantagens:

- Fácil integração entre diferentes plataformas
- Escalabilidade
- Reutilização de código e serviços
- Uso padronizado de verbos HTTP (GET, POST, PUT, DELETE)

4. Qual a função do npm no ecossistema Node.js? Explique como ele é usado no gerenciamento de dependências.

O **npm** (Node Package Manager) é o gerenciador de pacotes do Node.js.

Sua função é instalar, atualizar e remover bibliotecas (dependências) necessárias para o projeto.

Ele usa o arquivo package.json para registrar e organizar todas as dependências e versões usadas na aplicação.

5. Explique a diferença entre os métodos HTTP GET, POST, PUT e DELETE em APIs RESTful.

- **GET:** busca dados do servidor (sem alterar nada).
- **POST:** envia dados para criar um novo recurso.

- **PUT:** atualiza completamente um recurso existente.
- **DELETE:** remove um recurso do servidor.

6. Descreva o que é o Vue CLI e como ele facilita o desenvolvimento de projetos Vue.js.

O **Vue CLI** é uma ferramenta de linha de comando para criar e gerenciar projetos Vue.js.

Ele facilita o desenvolvimento ao gerar a estrutura do projeto automaticamente, configurar dependências, rodar servidor local e permitir personalização com plugins e presets.

7. Qual é o propósito do formato JSON em APIs RESTful? Por que ele é amplamente utilizado?

O JSON (JavaScript Object Notation) é usado para enviar e receber dados entre cliente e servidor de forma leve e estruturada.

É amplamente utilizado por ser fácil de ler, escrever e interpretar, além de ser nativamente compatível com JavaScript.

8. Compare o uso de Vue.js em uma aplicação front-end com o uso de bibliotecas como jQuery. Quais as vantagens do Vue.js?

jQuery manipula diretamente o DOM, sendo útil para tarefas simples, mas difícil de manter em projetos grandes.

Já o **Vue.js** usa componentes reativos e atualiza o DOM de forma eficiente.

Vantagens do Vue.js: melhor organização do código, reatividade, componentização, e integração mais fácil com frameworks modernos.

9. Cite dois cenários em que o PHP seria mais indicado que o Node.js para o desenvolvimento de um projeto.

1. **Sistemas baseados em WordPress ou CMSs prontos**, que já usam PHP nativamente.
2. **Hospedagens compartilhadas baratas**, onde o suporte a PHP é mais comum e simples de configurar.

10. Explique a relação entre Vue.js e o DOM reativo. Como isso melhora a performance de uma aplicação?

Vue.js utiliza um **DOM reativo**, o que significa que ele detecta automaticamente mudanças nos dados e atualiza apenas os elementos necessários na tela. Isso melhora a performance ao evitar recarregamentos completos do DOM, tornando as atualizações mais rápidas e eficientes.

11. Qual é a importância de separar o front-end (Vue.js) do back-end (Node.js ou PHP) no desenvolvimento de aplicações?

Separar front-end e back-end permite maior organização, escalabilidade e reutilização de código.

Cada parte foca em responsabilidades diferentes: o front-end cuida da interface do usuário, enquanto o back-end gerencia dados e regras de negócio. Isso facilita a manutenção, testes e uso de APIs em múltiplas plataformas (web, mobile, etc.).

12. O que é o comando `npm install` e como ele se diferencia de `npm install -g`?

`npm install` instala dependências locais do projeto, salvas no `node_modules` da pasta atual.

`npm install -g` instala pacotes globalmente, tornando-os acessíveis de qualquer lugar do sistema via terminal.

13. Por que o método POST em APIs RESTful é usado para criação de recursos? Dê um exemplo prático.

O método **POST** é usado para **criar** novos recursos porque ele envia dados ao servidor que ainda **não existem**.

Exemplo prático: enviar um formulário de cadastro para a rota `/usuarios` cria um novo usuário no banco de dados.

14. Explique o conceito de modularidade em APIs RESTful e como isso pode beneficiar um projeto grande.

Modularidade é dividir a API em partes menores e organizadas por função (ex: rotas de usuários, produtos, pedidos).

Isso facilita a **manutenção**, o **reaproveitamento de código** e o **trabalho em equipe**, além de tornar o projeto mais escalável e compreensível.

15. Qual o papel do yarn no ecossistema de desenvolvimento? Compare-o com o npm.

Yarn é um gerenciador de pacotes, assim como o npm, usado para instalar e gerenciar dependências de projetos JavaScript.

Comparação:

- Yarn é mais rápido em instalações repetidas (cache eficiente).
 - Usa o yarn.lock para garantir versões exatas das dependências.
 - npm é o padrão oficial do Node.js e tem ampla compatibilidade.
- Ambos cumprem funções similares, com pequenas diferenças de performance e controle de dependências.