

## **Parte 1: Questões Teóricas**

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

Vue.js é um framework progressivo para construção de interfaces de usuário, focado na camada de visualização. Suas principais características abrangem a reatividade de dados, que permite atualização automática do DOM, o uso de componentes reutilizáveis para modularizar a interface e a fácil integração com outras bibliotecas ou projetos já existentes.

**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 porque permite a execução de código JavaScript no lado do servidor, promovendo alta performance devido ao seu modelo assíncrono e orientado a eventos. Diferente de PHP, que é baseado em processos bloqueantes, o Node.js utiliza um loop de eventos não-bloqueante, tornando-o mais eficiente para aplicações em tempo real e que exigem escalabilidade.

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

APIs RESTful são interfaces que seguem os princípios do estilo arquitetural REST, utilizando métodos HTTP para manipulação de recursos. Suas principais vantagens incluem simplicidade na comunicação entre sistemas, escalabilidade, independência de linguagem e facilidade de integração com diferentes plataformas e serviços.

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

O npm é o gerenciador de pacotes oficial do Node.js, responsável por facilitar a instalação, atualização e remoção de bibliotecas necessárias para um projeto. Ele mantém um arquivo de configuração chamado package.json, que descreve todas as dependências, permitindo sua instalação automática e padronizada entre diferentes ambientes.

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

O método GET é usado para recuperar recursos, o POST para criar novos recursos, o PUT para atualizar recursos existentes e o DELETE para removê-los. Essa diferenciação permite que APIs RESTful mantenham uma organização semântica clara, facilitando a manutenção e a padronização na comunicação entre cliente e 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 que automatiza a configuração inicial e a manutenção de projetos Vue.js. Ele permite criar estruturas de pastas otimizadas, integrar plugins e realizar build e testes de maneira simplificada, poupando tempo e reduzindo a complexidade da configuração manual.

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

O JSON (JavaScript Object Notation) é utilizado para troca de dados estruturados de forma leve e legível entre cliente e servidor. Ele é amplamente adotado por sua simplicidade, compatibilidade nativa com JavaScript e facilidade de interpretação por diversas linguagens de programação.

**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?**

Enquanto o jQuery atua principalmente como uma biblioteca para manipulação direta do DOM, o Vue.js oferece uma abordagem declarativa com componentes reativos, que simplificam o desenvolvimento de interfaces dinâmicas. O Vue reduz a necessidade de manipulações manuais e facilita a manutenção de projetos grandes e complexos.

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

O PHP é mais indicado em projetos que demandam hospedagem compartilhada tradicional, onde já há suporte nativo para PHP, e em sistemas legados que utilizam CMSs como WordPress, amplamente baseados nessa linguagem. Nessas situações, o uso de PHP pode reduzir custos e facilitar a integração com ferramentas existentes.

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

O Vue.js utiliza um sistema reativo que observa alterações nos dados e atualiza automaticamente apenas as partes necessárias do DOM virtual, minimizando renderizações desnecessárias. Isso melhora a performance porque reduz o custo computacional e torna a interface mais responsiva e eficiente.

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

A separação entre front-end e back-end promove modularidade, permitindo que cada camada seja desenvolvida, testada e escalada de forma independente. Essa arquitetura facilita a manutenção, possibilita o uso de diferentes tecnologias em cada camada e favorece a reutilização de APIs por diferentes clientes, como aplicações web e móveis.

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

O comando npm install instala dependências de um projeto localmente na pasta node\_modules, enquanto npm install -g instala pacotes de forma global, tornando-os acessíveis em qualquer diretório do sistema. A escolha depende da necessidade de uso local para projetos específicos ou global para ferramentas de linha de comando.

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

O método POST é utilizado para criar recursos porque ele envia dados ao servidor sem a necessidade de um identificador prévio do recurso. Por exemplo, em uma aplicação de cadastro de usuários, ao enviar os dados de um novo usuário via POST, o servidor cria o registro e retorna um identificador único.

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

A modularidade em APIs RESTful consiste em dividir funcionalidades em endpoints específicos e bem definidos, cada um responsável por um conjunto limitado de operações. Isso beneficia projetos grandes ao facilitar a manutenção, permitir atualizações pontuais e favorecer a colaboração entre equipes distintas sem afetar todo o sistema.

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

O Yarn é um gerenciador de pacotes alternativo ao npm, criado para oferecer maior velocidade, segurança e controle sobre dependências. Ele utiliza cache local eficiente, suporta instalação paralela de pacotes e bloqueia versões de dependências de forma mais rigorosa, reduzindo problemas de inconsistência em ambientes diferentes.