

---

---

## Questões

---

---

### 1. Descreva as Principais Etapas no Ciclo de Vida de um componente no Vue.js e explique a importância de cada uma delas.

**Criação (Creation):** Nesta etapa, o componente é inicializado, as propriedades são configuradas e o estado inicial é estabelecido. É crucial para a configuração inicial do componente.

**Montagem (Mounting):** Aqui, o componente é conectado ao **DOM** e sua representação visual é renderizada. É onde ocorre a primeira atualização, importante para exibir o componente na interface do usuário.

**Atualização (Updating):** Quando o estado do componente muda, ele é re-renderizado, passando por essa etapa de atualização. É importante para refletir mudanças no estado e na interface do usuário.

**Destruição (Destruction):** O componente é removido do **DOM** e destruído, liberando recursos e removendo **Event Listeners**. Importante para evitar vazamentos de memória e manter a aplicação eficiente.

### 2. Quais são os passos básicos para integrar o Vue.js em um projeto de aplicação web (como incluir o framework para uso)?

**Inclusão do Vue.js via CDN ou Pacote NPM:** Adicionar a biblioteca **Vue.js** ao projeto.

**Criação de uma instância Vue:** Criar uma instância **Vue** para controlar uma parte específica da aplicação.

**Montagem da instância Vue:** Definir um elemento **HTML** onde a instância **Vue** será montada.

### 3. Explique como funciona o mecanismo de requisição e resposta com HTTP (GET POST PUT DELETE) utilizando JSON com VueJs.

Para realizar requisições **HTTP (GET, POST, PUT, DELETE)**, o **Vue.js** pode usar a biblioteca **axios** ou **fetch** para fazer as requisições.

Requisições são feitas usando métodos como **axios.get()**, **axios.post()**, **axios.put()** e **axios.delete()** para realizar operações **CRUD**.

Para lidar com as respostas, são utilizadas **promises** ou **async/await** para tratar os dados recebidos.

**JSON** é um formato de dados comumente usado para enviar e receber informações entre o **frontend (Vue.js)** e o **backend (servidor)**.

### 4. Explique a diferença entre o uso de Mustache e Diretivas e forneça um exemplo prático de uso para cada um.

**Mustache ({{ }})**: É usado para interpolar dados no **HTML**, exibindo o valor de uma variável na interface. Por exemplo:

```
<div>{{ mensagem }}</div>
```

**Diretivas**: São atributos especiais que iniciam com **v-** e são usados para aplicar comportamentos dinâmicos no **DOM**. Por exemplo:

```
<button v-on:click="incrementar">{{ contador }}</button>
```

Neste exemplo, **v-on:click** é uma diretiva que escuta o evento de clique e executa o método **incrementar**.

## 5. Assinale a alternativa que define corretamente o Vue.js e suas características principais.

- ☒ É um framework JavaScript para construir interfaces de usuário, com destaque para a reatividade e componentização.
- ☐ É uma linguagem de programação orientada a objetos com sintaxe semelhante ao JavaScript, voltada para interfaces de usuário.
- ☐ É um banco de dados não relacional amplamente utilizado para armazenar dados de aplicativos web.
- ☐ É um framework CSS para estilizar páginas web de maneira eficiente.

## 6. O que é um Componente no Vue.js?

- ☐ Um componente é uma função JavaScript que retorna uma interface de usuário.
- ☐ Um componente é um objeto JavaScript que armazena dados e funções relacionadas.
- ☒ Um componente é uma instância do Vue.js que pode ser reutilizada em seu aplicativo.
- ☐ Um componente é uma variável que armazena informações sobre a aplicação.

## 7. Como podem ser incorporadas **Condicionais** nos modelos Vue.js?

- ☒ Utilizando a diretiva v-if.
- ☐ Utilizando a diretiva v-for.
- ☐ Utilizando a diretiva v-else.
- ☐ Utilizando a diretiva v-switch.

## 8. Como **Eventos** podem ser utilizados no Vue.js?

- ☒ Utilizando a diretiva v-on.
- ☐ Utilizando a diretiva v-emit.
- ☐ Utilizando a diretiva v-bind.
- ☐ Utilizando a diretiva v-event.

## 9. Qual é a principal **Vantagem** de utilizar o Vue.js?

- ☐ Facilita a manipulação direta do DOM.
- ☐ Torna desnecessário o uso de HTML e CSS.
- ☒ Permite a criação de interfaces de usuário reativas e eficientes.
- ☐ Simplifica a implementação lógica do servidor.

## 10. Assinale a alternativa que explica a diretiva v-for e sua aplicação específica.

- ☒ A diretiva v-for é utilizada para criar loops em modelos Vue.js, sendo especialmente útil para renderizar listas de itens com base em um array.
- ☐ A diretiva v-for é utilizada para criar funções assíncronas em Vue.js, sendo especialmente útil para operações de I/O.
- ☐ A diretiva v-for é utilizada para criar condicionais em Vue.js, sendo especialmente útil para controlar a visibilidade de elementos no DOM.
- ☐ A diretiva v-for é utilizada para criar animações em Vue.js, sendo especialmente útil para tornar as transições de página mais suaves.