

Trilha 09

Avaliação Final da Trilha

Avaliação Final - Vue.js do Básico ao Avançado

Instruções:

- Leia atentamente cada questão.
 - Escolha apenas uma alternativa correta para cada questão.
 - Responda todas as questões antes de conferir o gabarito ao final do documento.
-

Questões Objetivas

- 1. O que é o Vue.js?**
 - a) Uma biblioteca para manipulação de banco de dados
 - b) Um framework JavaScript progressivo para construção de interfaces de usuário
 - c) Um compilador de código-fonte para otimização de desempenho
 - d) Uma linguagem de programação usada para aplicações mobile
 - e) Um banco de dados relacional utilizado em projetos Vue.js
- 2. Qual das alternativas abaixo não é uma vantagem do Vue.js?**
 - a) Facilidade de aprendizado
 - b) Gerenciamento eficiente de estado
 - c) Arquitetura baseada em componentes
 - d) Necessidade de múltiplas linguagens de programação para funcionar
 - e) Possibilidade de integração com outras bibliotecas e frameworks
- 3. O que é necessário para instalar o Vue CLI?**
 - a) Apenas o navegador Chrome
 - b) Apenas o Node.js e npm
 - c) O Vue.js instalado globalmente
 - d) Apache e MySQL
 - e) Nenhuma instalação adicional é necessária
- 4. Qual dos comandos abaixo cria um novo projeto Vue.js utilizando o Vue CLI?**
 - a) vue new project
 - b) npm install vue
 - c) vue create my-project
 - d) node init vue-project
 - e) vue setup project

5. **O que é uma API RESTful?**
 - a) Uma biblioteca de JavaScript para manipulação de estados
 - b) Um conjunto de regras para comunicação entre sistemas web utilizando padrões HTTP
 - c) Um framework para estilização de interfaces gráficas
 - d) Uma ferramenta exclusiva para integração de bancos de dados com Vue.js
 - e) Um formato de banco de dados relacional

6. **O que significa "Two-Way Data Binding" no Vue.js?**
 - a) A capacidade de um dado ser atualizado apenas no template
 - b) O ciclo de vida de um componente Vue
 - c) A sincronização automática entre o modelo de dados e a interface do usuário
 - d) A importação automática de módulos Vue
 - e) O processo de atualização do DOM manualmente pelo desenvolvedor

7. **Qual diretiva Vue.js é utilizada para vincular atributos HTML a variáveis do Vue?**
 - a) v-if
 - b) v-for
 - c) v-bind
 - d) v-model
 - e) v-slot

8. **Qual das opções abaixo é usada para percorrer uma lista no Vue.js?**
 - a) v-for
 - b) v-if
 - c) v-show
 - d) v-bind
 - e) v-model

9. **No Vue.js, qual evento pode ser utilizado para capturar um clique do usuário?**
 - a) @click
 - b) @keypress
 - c) @input
 - d) @mouseover
 - e) @change

10. **Qual diretiva Vue.js deve ser usada para controlar a exibição condicional de um elemento?**
 - a) v-show
 - b) v-if
 - c) v-for
 - d) v-bind
 - e) v-html

11. **O que são componentes no Vue.js?**
 - a) Funções reutilizáveis para estilização de elementos
 - b) Objetos reutilizáveis que encapsulam lógica, estrutura e estilos
 - c) Diretivas especiais para manipulação do DOM
 - d) Uma biblioteca externa para manipulação de dados
 - e) Ferramentas de backend para comunicação com APIs

12. **Qual diretiva Vue é usada para passar dados de um componente pai para um componente filho?**
 - a) v-bind
 - b) v-slot
 - c) v-on
 - d) v-for
 - e) v-model

13. **No Vue.js, como os componentes filhos podem se comunicar com os componentes pais?**
 - a) Através de eventos personalizados emitidos com \$emit()
 - b) Apenas utilizando propriedades (props)
 - c) Com v-model
 - d) Com v-slot
 - e) Não é possível comunicação entre componentes

14. **O que são slots em um componente Vue.js?**
 - a) Locais onde os componentes são renderizados dinamicamente
 - b) Elementos HTML que permitem a passagem de conteúdo dinâmico para componentes filhos
 - c) Eventos personalizados que disparam mudanças no estado
 - d) Métodos que substituem propriedades dentro de um componente
 - e) Métodos utilizados para estilização avançada no Vue.js

- 15. Qual a principal diferença entre Options API e Composition API no Vue.js?**
- a) Composition API permite organizar código de forma mais modular e reutilizável
 - b) Options API usa apenas JavaScript moderno e não suporta componentes reutilizáveis
 - c) Composition API só pode ser usada com Vuex
 - d) Options API é obrigatória no Vue 3
 - e) Não há diferença entre elas
- 16. O que é o Pinia no Vue.js?**
- a) Um ORM para manipular bancos de dados
 - b) Um gerenciador de estado global que substitui o Vuex
 - c) Um plugin para estilização de interfaces Vue.js
 - d) Um framework alternativo ao Vue.js
 - e) Uma ferramenta de teste unitário para Vue.js
- 17. Qual dessas opções define corretamente uma store no Pinia?**
- a) `defineStore('minhaStore', { state: () => ({ nome: 'João' }) })`
 - b) `createPinia({ nome: 'minhaStore' })`
 - c) `useStore({ nome: 'minhaStore', state: {} })`
 - d) `storeModule({ id: 'minhaStore' })`
 - e) `new PiniaStore('minhaStore')`
- 18. Como instalar o Pinia em um projeto Vue.js?**
- a) `npm install pinia`
 - b) `vue add pinia`
 - c) `npm install vuex-pinia`
 - d) `npm install vue-store`
 - e) `pinia install`
- 19. O que são Getters no Pinia?**
- a) Métodos que retornam dados processados a partir do estado da store
 - b) Ações assíncronas que fazem chamadas a APIs
 - c) Mutations responsáveis por atualizar o estado diretamente
 - d) Diretivas Vue.js usadas para gerenciar templates
 - e) Propriedades do Vue Router para definir rotas dinâmicas

20. Como acessar um estado armazenado no Pinia dentro de um componente Vue?

- a) Usando `useMinhaStore().meuEstado`
- b) Com `store.getState('meuEstado')`
- c) Usando `this.$state.meuEstado`
- d) Definindo `state` diretamente no `setup()`
- e) Não é possível acessar estados dentro de componentes

21. Qual a principal vantagem do Vuetify no desenvolvimento Vue.js?

- a) Permite o desenvolvimento de aplicações server-side
- b) Facilita a criação de interfaces utilizando Material Design
- c) É uma alternativa ao Vue Router
- d) Permite a compilação automática de código JavaScript
- e) Substitui a necessidade de usar Pinia

22. Como adicionar o Vuetify a um projeto Vue.js já existente?

- a) `npm install vuetify`
- b) `vue add vuetify`
- c) `npm install @vue/material`
- d) `vue add bootstrap`
- e) `npm install vue-router`

23. No Vuetify, qual componente é utilizado para criar botões estilizados?

- a) `<v-btn>`
- b) `<v-button>`
- c) `<v-bt>`
- d) `<vuetify-button>`
- e) `<button>`

24. Qual ferramenta pode ser usada para validar formulários com Vuetify?

- a) `rules`
- b) `v-validate`
- c) `v-model`
- d) `validate-form`
- e) `vuetify-validator`

25. Qual comando instala o Bootstrap Vue em um projeto Vue.js?
- a) `npm install bootstrap-vue`
 - b) `vue add bootstrap-vue`
 - c) `npm install bootstrap`
 - d) `vue-cli bootstrap`
 - e) `vue install bootstrap`
26. Qual das opções abaixo melhor descreve o Express.js?
- a) Um banco de dados utilizado no Vue.js
 - b) Um framework Node.js para criação de servidores e APIs RESTful
 - c) Uma biblioteca para estilização de componentes Vue
 - d) Um compilador para otimização de código Vue.js
 - e) Um ORM para manipulação de bancos de dados
27. Qual ORM é frequentemente utilizado para manipular bancos de dados MySQL no Node.js?
- a) Sequelize
 - b) Mongoose
 - c) Firebase
 - d) TypeORM
 - e) Prisma
28. O que significa CRUD no contexto de APIs RESTful?
- a) Criar, Renderizar, Utilizar e Deletar
 - b) Cliente, Requisição, Usuário e Dados
 - c) Create, Read, Update e Delete
 - d) Commit, Response, Update e Download
 - e) Classe, Repositório, Utilitário e Dependência
29. Qual das seguintes opções não é um método HTTP usado em APIs RESTful?
- a) GET
 - b) POST
 - c) DELETE
 - d) RENDER
 - e) PUT

30. **O que é necessário para consumir uma API no Vue.js utilizando Axios?**
- a) Importar a biblioteca Axios e realizar uma requisição HTTP usando `axios.get()`
 - b) Criar um componente Vue personalizado para API
 - c) Utilizar Vue Router para acessar os dados da API
 - d) Configurar um banco de dados local antes de chamar a API
 - e) Criar uma store Pinia antes de utilizar qualquer API
31. **Qual framework PHP é utilizado para criar APIs RESTful de forma simplificada?**
- a) Laravel
 - b) Symfony
 - c) Slim Framework
 - d) CodeIgniter
 - e) Zend Framework
32. **Qual a principal função do middleware em uma API RESTful no Slim Framework?**
- a) Gerenciar o banco de dados MySQL
 - b) Criar layouts responsivos para o front-end
 - c) Interceptar requisições para adicionar lógica de autenticação ou manipulação de dados
 - d) Substituir o Pinia na manipulação de estados
 - e) Criar rotas dinâmicas sem necessidade de um servidor
33. **O que é um token JWT?**
- a) Um identificador único para cada usuário dentro do Pinia
 - b) Uma ferramenta de autenticação baseada em JSON para segurança em APIs
 - c) Um arquivo de configuração usado para definir variáveis globais
 - d) Um plugin do Vue CLI para melhorar o desempenho da aplicação
 - e) Um comando para compilar código Vue.js automaticamente
34. **Qual é o comando correto para instalar o Composer no ambiente de desenvolvimento PHP?**
- a) `npm install composer`
 - b) `apt-get install composer`
 - c) `curl -sS https://getcomposer.org/installer | php`
 - d) `php install composer`
 - e) `vue add composer`

35. O que o comando `php -S localhost:8000 -t public` faz ao rodar um projeto Slim Framework?
- a) Inicia um servidor local na porta 8000 servindo arquivos da pasta public
 - b) Cria um banco de dados MySQL automaticamente
 - c) Instala dependências do projeto
 - d) Executa uma compilação do código Vue.js
 - e) Substitui a store Pinia por uma API externa
36. Qual a principal vantagem do uso de Axios para consumir APIs no Vue.js?
- a) Funciona apenas com APIs RESTful
 - b) Permite requisições assíncronas de forma mais intuitiva do que a Fetch API
 - c) Funciona apenas em projetos Vue CLI
 - d) Exige autenticação JWT para qualquer requisição
 - e) Não possui suporte a métodos HTTP como DELETE e PUT
37. Qual API pode ser utilizada para buscar endereços a partir de um CEP?
- a) OpenWeatherMap
 - b) ViaCEP
 - c) Firebase
 - d) Google Maps
 - e) Stripe API
38. No Vue.js, como pode-se exibir uma mensagem amigável caso ocorra um erro ao consumir uma API?
- a) Monitorando erros com `.catch()` ou `try...catch`
 - b) Criando um novo banco de dados para armazenar logs
 - c) Reiniciando o servidor Vue.js sempre que um erro ocorre
 - d) Evitando chamadas de API dentro de componentes Vue
 - e) Desativando a store Pinia para evitar conflitos
39. Qual cabeçalho HTTP precisa ser configurado no back-end para permitir chamadas de uma API em domínios diferentes?
- a) Content-Type
 - b) Authorization
 - c) Access-Control-Allow-Origin
 - d) X-Requested-With
 - e) Cache-Control

40. **Qual das seguintes opções não é um erro comum ao consumir APIs no Vue.js?**

- a) Erro de CORS
- b) Tempo limite (Timeout) excedido
- c) Resposta em formato JSON inválido
- d) Falha ao acessar o Vue Router
- e) Falha na autenticação JWT