

b) Um framework JavaScript progressivo para construção de interfaces de usuário

questão 2:

**d) Necessidade de múltiplas linguagens de programação para funcionar**

questão 3:

**b) Apenas o Node.js e npm**

questão 4:

**c) vue create my-project**

questão 5:

**b) Um conjunto de regras para comunicação entre sistemas web utilizando padrões HTTP**

questão 6:

**c) A sincronização automática entre o modelo de dados e a interface do usuário**

questão 7:

**c) v-bind**

questão 8:

**a) v-for**

questão 9:

**a) @click**

questão 10:

**b) v-if**

questão 11:

**b) Objetos reutilizáveis que encapsulam lógica, estrutura e estilos**

questão 12:

**a) v-bind**

questão 13:

**a) Através de eventos personalizados emitidos com \$emit()**

questão 14:

**b) Elementos HTML que permitem a passagem de conteúdo dinâmico para componentes filhos**

questão 15:

**a) Composition API permite organizar código de forma mais modular e reutilizável**

questão 16:

**b) Um gerenciador de estado global que substitui o Vuex**

questão 17:

**a) defineStore('minhaStore', { state: () => ({ nome: 'João' }) })**

questão 18:

**a) npm install pinia**

questão 19:

**a) Métodos que retornam dados processados a partir do estado da store**

questão 20:

**a) Usando useMinhaStore().meuEstado**

questão 21:

**b) Facilita a criação de interfaces utilizando Material Design**

questão 22:

**b) vue add vuetify**

questão 23:

**a) <v-btn>**

questão 24:

**a) rules**

questão 25:

**a) npm install bootstrap-vue**

questão 26:

**b) Um framework Node.js para criação de servidores e APIs RESTful**

questão 27:

**a) Sequelize**

questão 28:

**c) Create, Read, Update e Delete**

questão 29:

**d) RENDER**

questão 30:

**a) Importar a biblioteca Axios e realizar uma requisição HTTP usando axios.get()**

questão 31:

**c) Slim Framework**

questão 32:

**c) Interceptar requisições para adicionar lógica de autenticação ou manipulação de dados**

questão 33:

**b) Uma ferramenta de autenticação baseada em JSON para segurança em APIs**

questão 34:

**c) curl -sS <https://getcomposer.org/installer> | php**

questão 35:

**a) Inicia um servidor local na porta 8000 servindo arquivos da pasta public**

questão 36:

**b) Permite requisições assíncronas de forma mais intuitiva do que a Fetch API**

questão 37:

**b) ViaCEP**

questão 38:

**a) Monitorando erros com .catch() ou try...catch**

questão 39:

**c) Access-Control-Allow-Origin**

questão 40:

**d) Falha ao acessar o Vue Router**