

1. O que é o Vuetify e quais são suas principais vantagens ao desenvolver interfaces no Vue.js?

Vuetify é um framework de componentes visuais baseado em **Material Design**, feito especialmente para Vue.js.

Principais vantagens:

- Componentes prontos e responsivos (botões, cards, grids, etc.).
- Design consistente e profissional com pouco esforço.
- Integração fácil com Vue CLI.
- Agilidade no desenvolvimento de interfaces modernas.

2. Explique o processo de configuração inicial do Vuetify em um projeto Vue.js. Quais são os passos necessários para adicioná-lo ao projeto?

Passos para configurar o Vuetify:

Crie o projeto com Vue CLI (ou use um existente):

```
vue create meu-projeto
```

Acesse a pasta do projeto:

```
cd meu-projeto
```

Instale o Vuetify com Vue CLI:

```
vue add vuetify
```

🔍 Escolha um preset (padrão ou customizado) durante a instalação.

🔍 Pronto! O Vuetify estará configurado com layout base, tema e componentes prontos para uso.

3. Descreva o funcionamento do sistema de Grid do Vuetify. Como ele se compara ao sistema tradicional de CSS Flexbox e Bootstrap?

O **sistema de Grid do Vuetify** é baseado no **Flexbox** e utiliza uma estrutura de **12 colunas** responsivas. Ele usa componentes como:

- `<v-container>` (área de layout),
- `<v-row>` (linha),
- `<v-col>` (coluna).

4. Quais são os principais componentes do Vuetify? Escolha pelo menos três e explique suas funcionalidades e usos.

`<v-btn>` (Button)

Cria botões estilizados com Material Design. Suporta ícones, cores, tamanhos e eventos.

Uso: Ações como salvar, enviar, cancelar etc.

`<v-card>` (Card)

Componente para exibir conteúdo em blocos elegantes com título, texto, imagem e ações.

Uso: Apresentar informações organizadas, como perfis, produtos, notificações.

`<v-text-field>` (Campo de Texto)

Campo de entrada de dados com validação, rótulo, ícones e máscaras.

Uso: Formulários, login, buscas, etc.

5. Explique a diferença entre os componentes `v-btn`, `v-card` e `v-dialog`. Em quais situações cada um deles é mais adequado?

- **`<v-btn>` (Botão):**
Componente para **executar ações** (ex: salvar, excluir, abrir modais).
Mais adequado para: interações diretas do usuário.
- **`<v-card>` (Cartão):**
Bloco visual para **exibir conteúdo estruturado**, como textos, imagens, títulos e botões.
Mais adequado para: mostrar informações organizadas como produtos, perfis ou notícias.
- **`<v-dialog>` (Diálogo/Modal):**
Janela flutuante para **exibir conteúdo sobreposto** à interface principal, como alertas, formulários ou confirmações.
Mais adequado para: interações que exigem atenção ou entrada de dados sem sair da tela.

6. Como funciona a validação de formulários no Vuetify? Qual a importância do atributo rules ao definir um campo de entrada de dados?

A validação no Vuetify funciona com o atributo rules, que recebe um array de funções de validação. Cada função retorna true (válido) ou uma mensagem de erro (inválido).

Importância do rules:

- Permite validar campos de forma declarativa.
- Exibe automaticamente mensagens de erro no campo.
- Garante que os dados inseridos pelo usuário estejam corretos antes de enviar o formulário.

7. Explique como funcionam as Data Tables no Vuetify e como elas podem ser utilizadas para exibir grandes volumes de dados

As **Data Tables** (<v-data-table>) do Vuetify são componentes poderosos para exibir e organizar grandes volumes de dados em formato de tabela.

Funcionamento:

- Recebem os dados via :items e os cabeçalhos via :headers.
- Suportam recursos como **paginação, filtro, ordenação, slots personalizados e ações por linha**.
- Podem ser combinadas com APIs ou dados locais.

Uso comum: exibir listas de usuários, produtos, relatórios ou registros de banco de dados de forma interativa e responsiva.

8. O Vuetify permite a personalização avançada da interface. Explique como criar e configurar temas customizados.

O Vuetify permite definir **temas customizados** com cores personalizadas para componentes como botões, textos e barras.

Como configurar:

1. Acesse o arquivo de configuração do Vuetify (geralmente vuetify.js ou vuetify.ts).
2. Defina um tema em theme.themes.light ou dark.

```
export default createVuetify({  
  
  theme: {  
  
    themes: {  
  
      light: {  
  
        primary: '#1976D2',  
  
        secondary: '#424242',  
  
        accent: '#82B1FF',  
  
        error: '#FF5252',  
  
        success: '#4CAF50'  
  
      }  
  
    }  
  
  }  
  
})
```

9. Qual é a importância do uso de SCSS no Vuetify? Como ele pode ser utilizado para modificar estilos globalmente?

O **SCSS** (Sass) é importante no Vuetify porque permite usar **variáveis**, **aninhamentos e funções** para criar estilos mais organizados e reutilizáveis.

Vantagens no Vuetify:

- Personalização avançada de temas.
- Sobrescrita de estilos padrão de forma limpa.
- Criação de estilos globais com mais controle.

10. Como integrar APIs com Vuetify para preencher dinamicamente tabelas e formulários? Cite um exemplo de requisição e consumo de dados.

A integração de APIs com Vuetify é feita utilizando bibliotecas como axios para buscar dados externos. Esses dados são armazenados em variáveis reativas e usados em componentes como `<v-data-table>` para exibir listas ou `<v-text-field>` com v-model para formulários.

Por exemplo, ao carregar dados de usuários via API, os resultados podem ser atribuídos à propriedade `items` da tabela. Em formulários, dados recebidos ou enviados podem ser vinculados a campos de entrada. Isso torna a aplicação dinâmica, interativa e conectada com o backend.