**Questões Teóricas**

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

O Vuetify é um framework de componentes baseado em Material Design para Vue.js. Ele facilita o desenvolvimento de interfaces modernas e responsivas, fornecendo uma vasta coleção de componentes prontos, como botões, cards, formulários e tabelas. Suas principais vantagens incluem a consistência visual, a economia de tempo no desenvolvimento e a alta personalização.

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

O processo começa com a criação de um projeto Vue.js utilizando Vue CLI. Após isso, o Vuetify pode ser instalado via npm ou yarn, e configurado automaticamente com o comando vue add vuetify. Esse comando realiza as alterações necessárias no projeto, como a inclusão do Vuetify no arquivo main.js e a configuração de temas e estilos padrões.

**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 inspirado no grid system do Bootstrap. Ele organiza o layout em linhas (v-row) e colunas (v-col), permitindo criar interfaces responsivas de forma intuitiva. A vantagem é a integração nativa com os componentes Vuetify, evitando a necessidade de escrever muitas classes CSS personalizadas.

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

* **v-btn:** botão altamente personalizável, usado para ações como salvar, enviar ou navegar.
* **v-card:** componente para exibir informações em blocos organizados, muito utilizado em dashboards e listagens.
* **v-dialog:** componente para criar caixas de diálogo modais, úteis para formulários rápidos, confirmações e mensagens.

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

O v-btn é utilizado para ações diretas, como cliques em formulários ou navegação. O v-card é ideal para exibir informações agrupadas de forma organizada e visualmente agradável. Já o v-dialog é usado em situações em que se deseja exibir informações temporárias ou solicitar interações sem sair da página atual.

**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 é feita utilizando o atributo rules, que recebe um array de funções responsáveis por validar os dados inseridos. Cada função retorna true quando a validação é bem-sucedida ou uma mensagem de erro quando falha. Esse sistema garante que apenas dados válidos sejam enviados, aumentando a confiabilidade da aplicação.

**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) permitem exibir dados em formato tabular, com suporte a paginação, ordenação e filtros. Elas são ideais para exibir grandes conjuntos de informações de forma organizada e interativa, podendo ser preenchidas dinamicamente com dados vindos de APIs.

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

A personalização de temas pode ser feita configurando o objeto theme ao inicializar o Vuetify no projeto. É possível definir cores primárias, secundárias, de erro, sucesso, entre outras. Além disso, pode-se alternar entre tema claro e escuro, ou até mesmo criar múltiplos temas para diferentes contextos da aplicação.

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

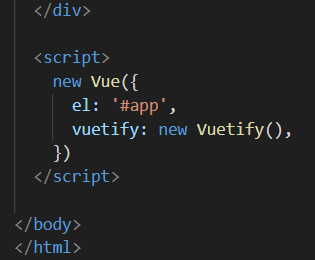
O SCSS permite modificar variáveis de estilo globais, como cores e espaçamentos, antes mesmo da compilação do projeto. Isso garante maior flexibilidade e consistência visual em toda a aplicação, já que mudanças podem ser aplicadas em larga escala de forma centralizada.

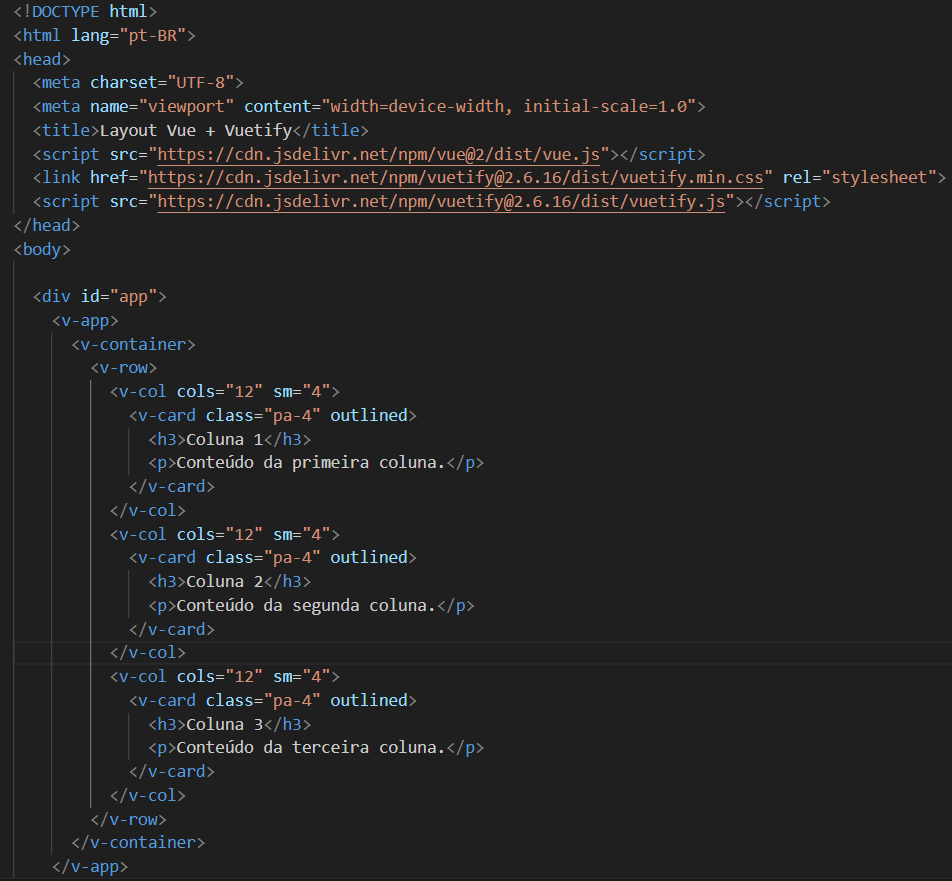
**10. Como integrar APIs com Vuetify para preencher dinamicamente tabelas e formulários? Cite um exemplo de requisição e consumo de dados.**

A integração pode ser feita utilizando bibliotecas como Axios ou Fetch API. Por exemplo, uma requisição GET pode buscar dados de uma API e preencher uma v-data-table. Da mesma forma, formulários podem ser preenchidos com dados vindos do servidor e enviados novamente através de requisições POST ou PUT.

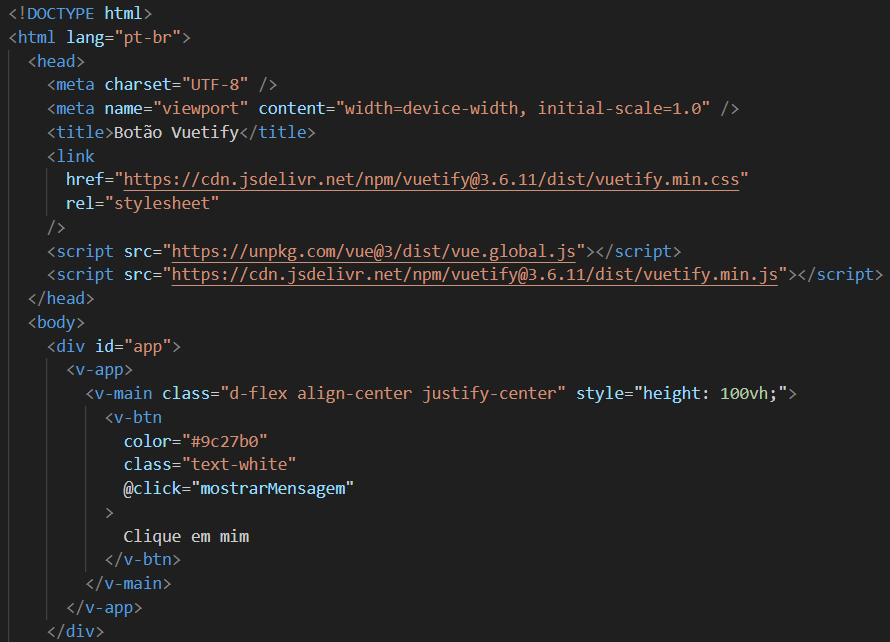
**Questões Práticas**

**1. Crie um projeto Vue.js com Vuetify e implemente um layout básico utilizando o sistema de Grid. O layout deve conter três colunas que se ajustam para telas pequenas.**

****

****

**2. Crie um botão (v-btn) estilizado com uma cor personalizada e um evento de clique que exibe uma mensagem no console.**

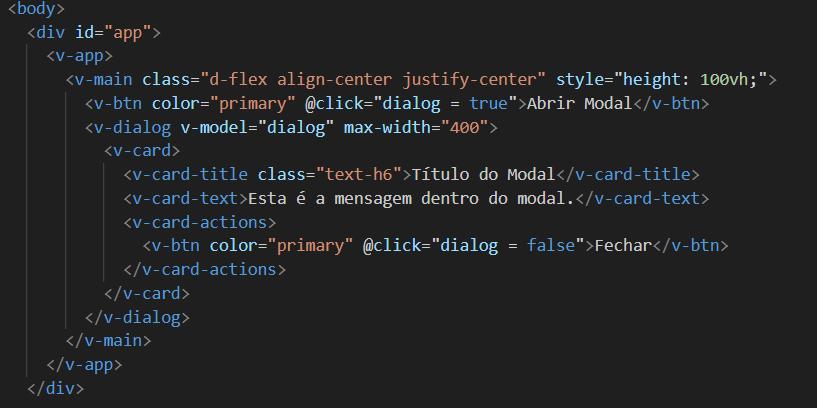
****



**3. Implemente um v-card contendo uma imagem, um título e um botão que exibe um alerta ao ser pressionado.**

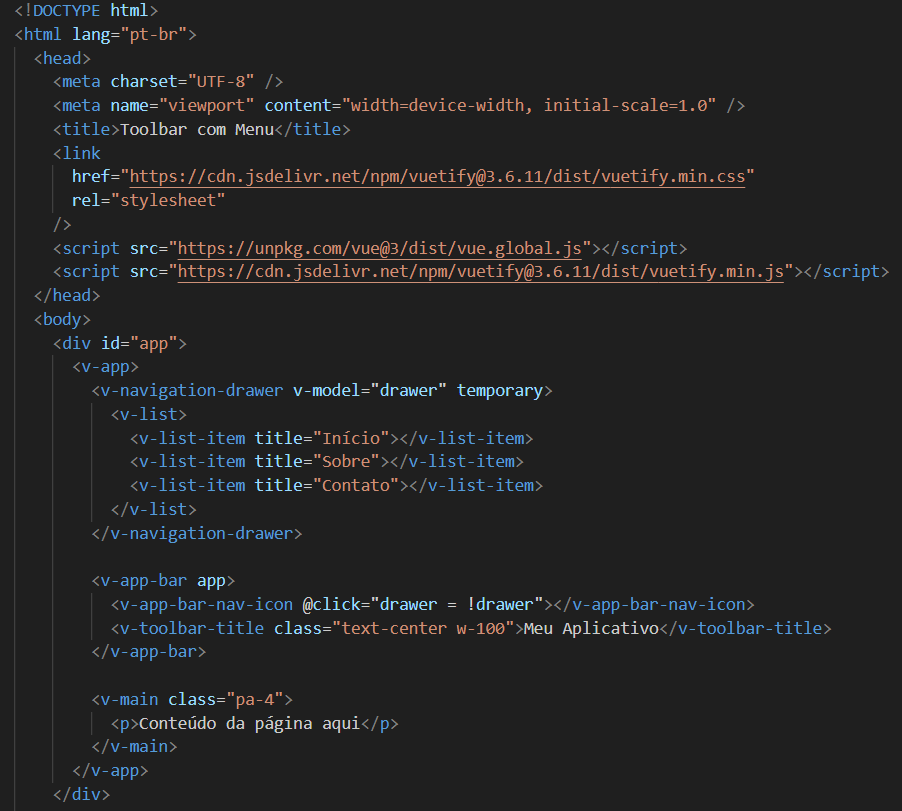
****

**4. Crie um modal (v-dialog) que seja ativado ao clicar em um botão. O modal deve conter um título, uma mensagem e um botão para fechá-lo.**



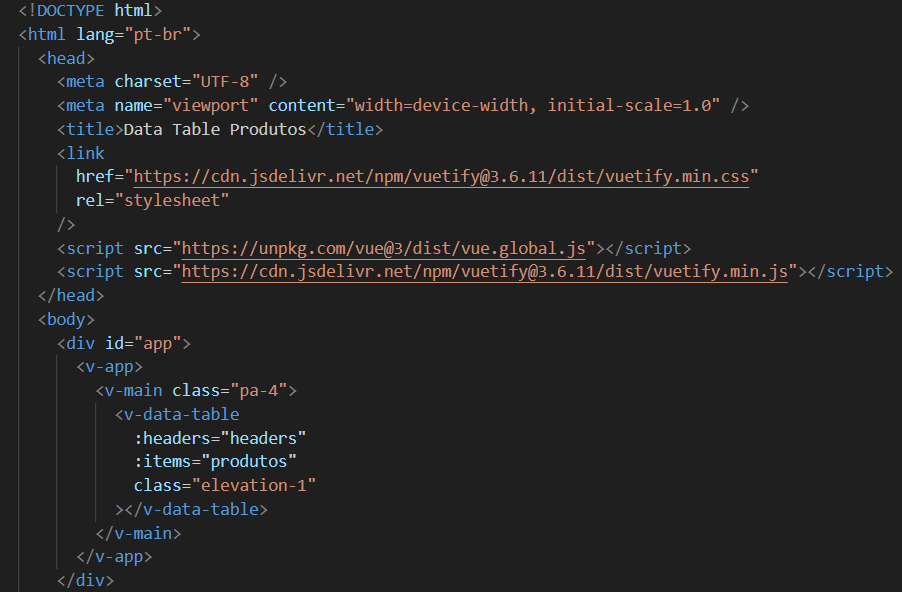


**5. Desenvolva um v-toolbar fixo na parte superior da tela que contenha um menu lateral (v-navigation-drawer) e um título centralizado.**



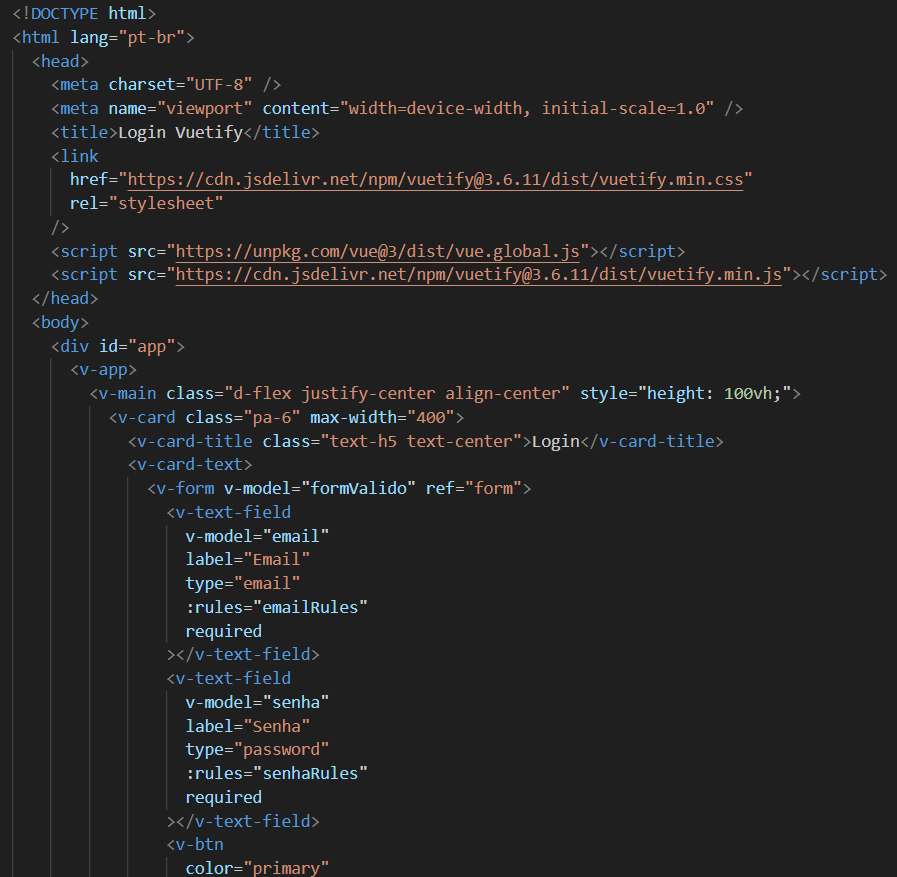


**6. Crie uma Data Table (v-data-table) que carregue uma lista de produtos a partir de um array estático no data(), contendo colunas para nome, preço e estoque.**

****

****

**7. Implemente um formulário de login utilizando Vuetify, com validação de campos para email e senha. Os campos devem exibir mensagens de erro quando não preenchidos corretamente.**

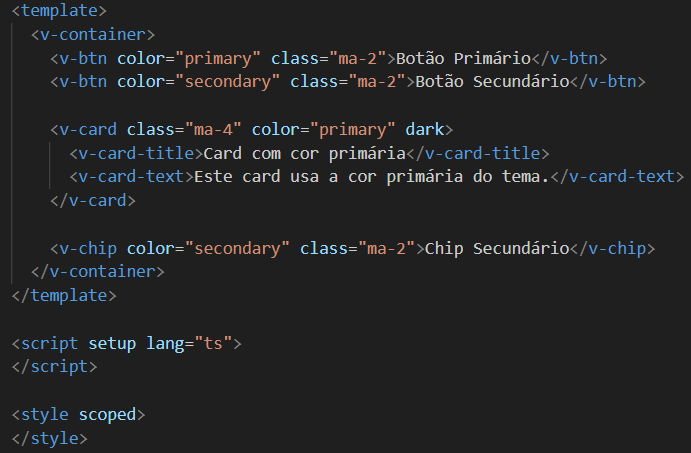
****

****

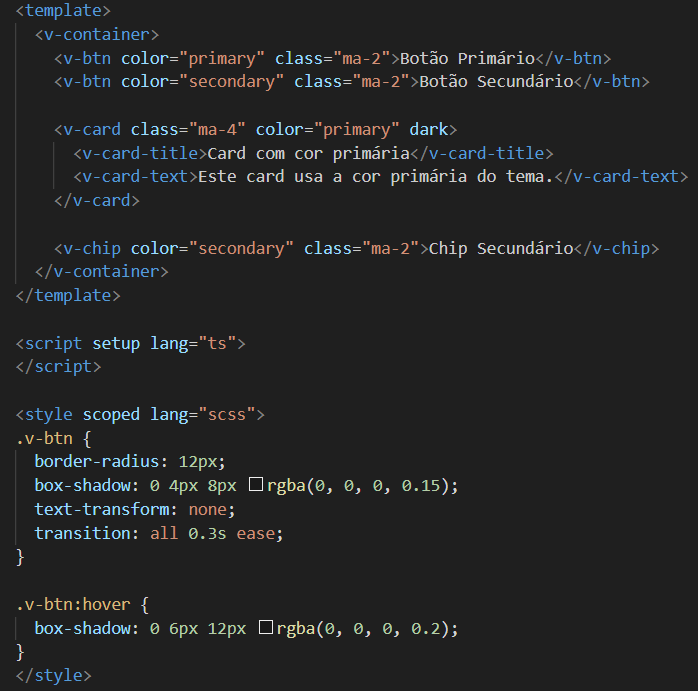
****

**8. Crie um tema customizado no Vuetify, alterando a cor primária para verde e a cor secundária para amarelo. Aplique essas cores nos componentes da interface.**

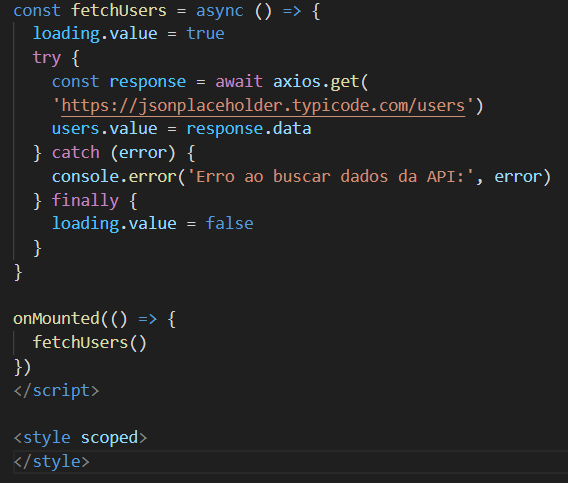
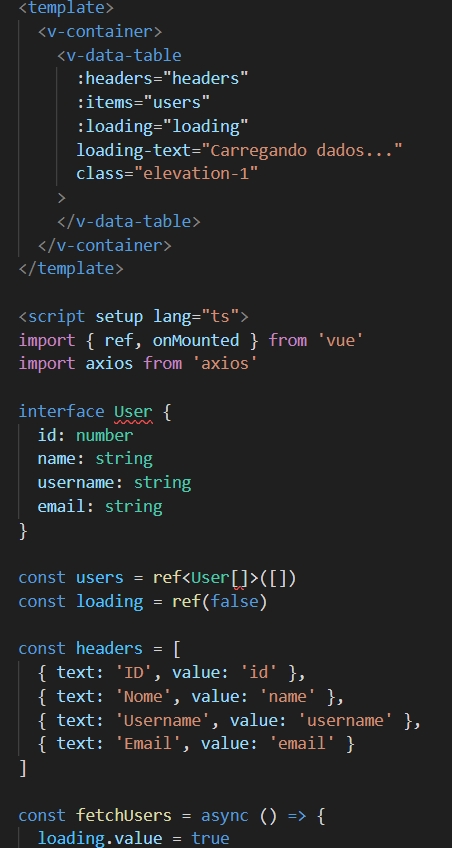
vuetify.js

Vue

**9. Utilizando SCSS, modifique a aparência dos botões no Vuetify para que fiquem arredondados e com uma sombra suave.**

****

**10. Faça a integração de uma API externa com Vuetify. Os dados da API devem ser exibidos dinamicamente em uma Data Table, atualizando os registros ao carregar a página.**

****