Mapa da Estrutura do Vue.js (Single File Component)

No Vue.js, a forma mais comum de estruturar um componente é usando o formato de Single File Component (SFC). Um arquivo .vue é dividido em blocos principais: <template>, <script> e <style>. Cada um tem uma responsabilidade clara na organização do código.

1) Seção <template>

Responsável por definir o HTML do componente. Aqui usamos diretivas do Vue (como v-if, v-for, v-model) para criar elementos dinâmicos.
Exemplo:

```
<template>
  <div>
    <h1>{{ titulo }}</h1>
    <p v-if="visivel">Esse texto aparece quando 'visivel' é true</p>
  </div>
</template>
```

2) Seção <script> (Options API)

Aqui definimos a lógica do componente usando a Options API. Exportamos um objeto com propriedades como data, methods, computed e watch.
Exemplo:

```
<script>
export default {
  name: "MeuComponente",
  data() {
    return {
      titulo: "Olá Vue!",
      visivel: true
    };
  },
  methods: {
    alternar() {
      this.visivel = !this.visivel;
    }
  }
};
</script>
```

3) Seção <script setup> (Composition API)

Com a Composition API, a sintaxe fica mais simples e moderna. Usamos ref e reactive para criar estados reativos.
Exemplo:

```
<script setup>
import { ref } from "vue";

const titulo = ref("Olá Vue 3!");
const visivel = ref(true);

function alternar() {
  visivel.value = !visivel.value;
}
</script>
```

4) Seção <style>

Usada para aplicar CSS ao componente. Quando usamos o atributo scoped, os estilos só afetam este componente.
Exemplo:

```
<style scoped>
h1 {
  color: blue;
}
</style>
```

Resumo do ciclo de um componente Vue:

- <template> → Estrutura visual (HTML). - <script> ou <script setup> → Lógica e estado (JavaScript). - <style> → Estilos (CSS). Essa separação ajuda a manter o código organizado e fácil de entender.