

GUIA VUE 3

Guia com explicações e exemplos de uso do Framework VUE.js 3, obtidos através do “Curso de Vue 3” do canal “Matheus Battisti - Hora de Codar”

AULA 02 – INSTALAÇÃO DO VUE

Formas de Instalar o Vue:

1. CDN

Instalação rápida via link de script. Ideal para testes, estudos ou projetos pequenos.

Etapas:

1) Inserir o script no <head> do HTML:

```
<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
```

2) Criar um elemento HTML com id, por exemplo:

```
<div id="app"></div>
```

3) Criar o script JavaScript com o Vue App:

```
const MyNameApp = {  
  data() {  
    return {  
      name: "Lucas",  
      age: 30  
    }  
  }  
}
```

```
Vue.createApp(MyNameApp).mount("#app");
```

Observações:

- Vue.createApp(objeto) → cria a aplicação Vue com as propriedades fornecidas.
- .mount("#app") → conecta a aplicação à <div> com id="app".
- data() precisa **ser uma função** que retorna um **objeto**, pois cada instância do Vue deve ter seu próprio estado isolado.

2. CLI (Command Line Interface)

Utilizada com Node.js, gera automaticamente a estrutura do projeto com suporte completo ao Vue.

Vantagens do CLI:

- Uso de ferramentas como Vue Router, Vuex e suporte a Single File Components (.vue).
- Mais escalável e profissional.
- Separação de arquivos e organização automatizada.

AULA 04 – UTILIZADO O VUE CLI

PASSOS:

1) Abrir o terminal e digitar : `npm install -g @vue/cli` (caso nunca tenha feito)

Isso instala globalmente o VUE no computador do usuário, caso utilize **npm install @vue/cli** em uma pasta sem o **-g**, o vue so estará instalado naquele local/projeto.

É necessário q pelo menos a versão v4.5 esteja instalada, caso não, utilizar **vue upgrade --next** para atualizar

2) Criar o projeto com `vue create` (nome do projeto)

2.1) Para criar o projeto com o VUE CLI, basta digitar o comando:

`vue create (nome_do_projeto)`

Exemplo: `vue create projeto_inicial_teste`

OBS: o nome do projeto só pode possuir letras minúsculas

2.2) Escolha de opções

Assim q rodar o comando de criação, o vue apresentará opções de projeto para criar, sendo: VUE 2 versão padrão, VUE 3 versão padrão, e a opção de selecionar manualmente as características do projeto VUE (como versão do vue, integração com TypeScript, rotas etc).

2.3) Escolher a versão VUE

Agora, o vue perguntará qual versão você deseja usar no projeto, tendo como opções as versões 2 ou 3 do VUE.

2.4) Escolher arquivo com as configurações

Agora o vue pede que escolha em qual arquivo serem inseridas as configurações do projeto, podendo ser no `package.json` ou um arquivo de configuração dedicado.

2.5) Escolha de pré definição

Por fim, ele perguntará se deve manter as configurações escolhidas neste projeto como pré-definição para a criação de projetos futuros. É aconselhável digitar N (apertar enter sem escolher nada seleciona a opção N) para q em projetos futuros suas opções possam ser customizadas e adaptadas para o contexto do projeto.

AULA 05 - COMPONENTES

São super importantes, onde cada componente pode integrar uma parte/segmento do site, desempenhando sua função própria. Pode, por exemplo, ser o rodapé, formulário, menu de navegação do site, etc

1) Tag <template>

É nesta tag q todo o conteúdo do componente vue aparecerá na tela

2) Exportando o componente

É necessário utilizar o script para exportar o componente, ou seja, para permitir q os demais componentes tenham acesso a ele.

Ex:

```
<script>

  export default{ //Exportando o componente como o Nome 'App'
    name: 'App'
  }

</script>
```

3) Importando um componente

Para utilizar um componente dentro de outro, é necessário importa-lo dentro do componente que irá o utilizar.

Ex:

```
import PrimeiroComponente from './components/PrimeiroComponente.vue'
```

Também é necessário que esse componente seja listado na lista de componentes externos presentes neste componente

Ex:

```
export default{ //Exportando o componente como o Nome 'App'
  name: 'App',
  components:{ //listando os componentes externos
    PrimeiroComponente
  }
}
```

3) Utilizando o componente

Basta criar uma tag com o nome do componente para utilizá-lo.

Ex:

```
<template>

  <PrimeiroComponente/> <!-- Aqui o componente é utilizado -->

</template>
```

OBS: O componente também pode ser utilizado como (embora não muito usual):

```
<template>

  <PrimeiroComponente> <!-- Aqui o componente é utilizado -->
  </PrimeiroComponente>

</template>
```

AULA 05 – DADOS EM COMPONENTES

Componentes podem conter dados, podendo inicializar com algum valor, e também possível de ser modificado durante a execução do programa. Os dados ficam em uma função chamada **data**, e esta função possui notação de função, pois deve retornar os dados.

Esta função deve retornar os dados em **formato de objeto**.

Ex:

```
export default{
  name: 'PrimeiroComponente',
  data(){ //função que manipula os dados do componente
    return{ //retorno dos dados do componente
      nome: 'Lucas' //dado definido
    }
  }
}
```

Utilizando o dado no componente:

```
<template>
  <div>
    <h1>Olá Vue!</h1>
    <p>Meu nome é {{ nome }}</p>
  </div>
</template>
```

OBS: O VUE só aceita um elemento base, ou seja, uma <div>, um <h1>, um <p> etc.

Por isso, é recomendado sempre envolver o conteúdo do componente em uma <div> pai no código, como no exemplo acima.

OBS 2: Um componente pode utilizar os dados de outro componente

Ex:

```
<template>
  <div>

    <PrimeiroComponente/> <!-- Aqui o outro componente é utilizado -->
    <p>O dado de app é: {{ testando }}</p> <!--dado deste componente -->

  </div>
</template>
```