

AULA 9: NAVEGAÇÃO E SPA COM VUE

Disciplina: FRONT-END FRAMEWORKS

Professor: André Neto



PORTAL DE EVENTOS VUE.JS

O PROJETO DA AULA

Você desenvolverá um **portal de eventos COP-30 Belém** do zero, construindo funcionalidades a cada etapa:

01

ESTILIZAÇÃO COM BOOTSTRAP

Design profissional e responsivo

02

DIRETIVAS VUE

Controle dinâmico da interface

03

NAVEGAÇÃO SPA

Vue Router para múltiplas páginas

04

INTEGRAÇÃO API

Dados dinâmicos em tempo real



Cada conceito inclui exemplo prático + tempo para aplicação

BOOTSTRAP VIA CDN

Bootstrap fornece componentes estilizados e responsivos sem esforço.

Como adicionar:

- Encontre os links CDN no site <https://getbootstrap.com/>
- Abra seu `index.html`
- Cole os links CDN no `<head>` e `<body>`

📄 🔗 Link do exemplo: <https://stackblitz.com/edit/projeto-aula-spa?file=index.html>

🔗 Teste do Bootstrap: <https://stackblitz.com/edit/projeto-aula-spa?file=src%2FApp.vue>



UTILIZANDO PARÂMETROS COM PROPS



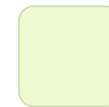
PROPS

parâmetros
enviados do pai
(App.vue) para o
filho
(Mensagem.vue).



DEFINEPROPS

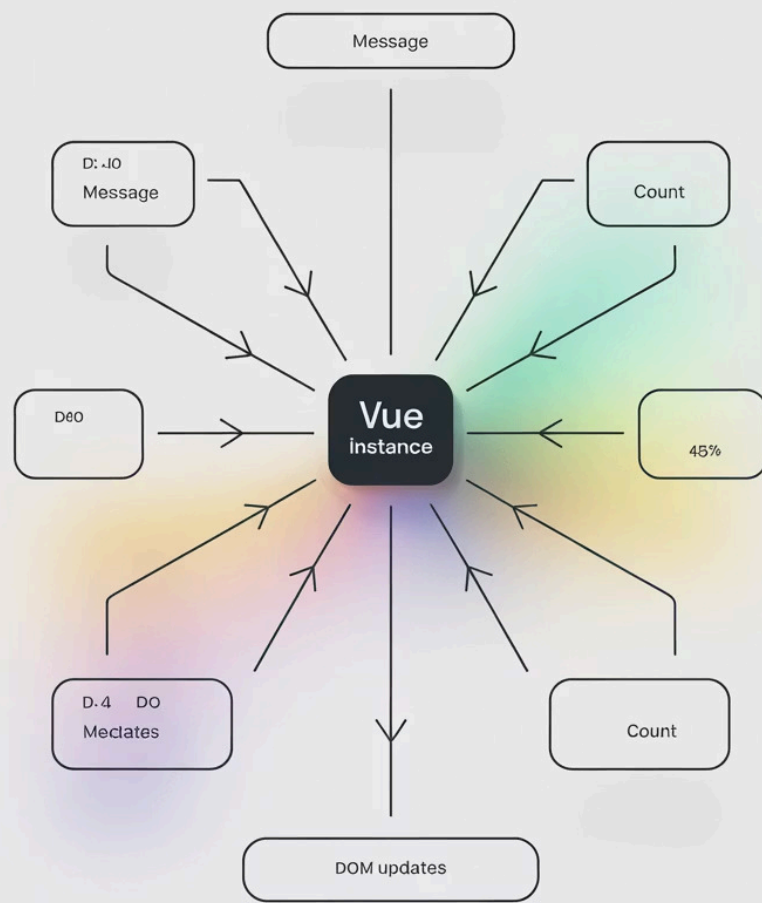
define quais props
o componente
aceita.



NO USO (APP.VUE)

passamos valores
como atributos do
componente.

Reactive Data Flow



EXEMPLO SIMPLES:

```
<script setup lang="ts">  
import { ref } from "vue";  
const mensagem = ref("Olá Vue!");  
</script>
```

```
<template>  
  <p>{{ mensagem }}</p>  
</template>
```

Se mensagem.value mudar → o <p> atualiza sozinho.

DIRETIVAS VUE: SEU KIT DE FERRAMENTAS

As diretivas são comandos especiais que conectam seus dados ao template HTML:

Diretiva	Função Principal
v-if / v-else	Exibir ou ocultar elementos conforme condição
v-for	Repetir elementos com base em listas
v-bind:atributo ou :atributo	Ligar atributos a dados dinâmicos
v-on:evento ou @evento	Escutar eventos de usuário
v-model	Ligação bidirecional com inputs

EXEMPLO 1: COMPONENTE BOTAO

```
1  <script setup>
2  import { ref } from 'vue';
3
4  const mostrar = ref(false);
5  ✓ function resetar() {
6    |   mostrar.value = false;
7    | }
8  </script>
```

/src/components/Botao.vue

ATIVIDADE PRÁTICA 1: BOTÃO COM PROPS

Agora é a sua vez! Aplique o conceito de **props** para tornar o componente **Botão** ainda mais flexível:

INTEGRAÇÃO

No seu ``App.vue`` ou em um novo componente, importe e utilize o ``Botao.vue``.

DEFINIR PROPRIEDADE

Crie duas novas `props` no componente: uma chamada `mensagem` onde o texto do alerta pode ser passado e uma



Dica: Lembre-se de como o `defineProps` funciona e como os atributos são passados para o componente filho!



ATIVIDADE PRÁTICA 2

SISTEMA DE CURTIDAS

1

CRIE DOIS BOTÕES

"Curtir" e "Não Curtir" com classes Bootstrap

2

USE @CLICK

Cada botão executa uma função diferente

3

MOSTRE MENSAGENS

Use `v-if` e `v-else` para exibir feedback

4

CONTE OS CLIQUES

Variável reativa `ref()` para armazenar contagem

EXEMPLO 2: COMPONENTE PALESTRANTES

```
1  <script setup lang="ts">
2  defineProps<{
3    participantes: Array;
4  }>();
5  </script>
```

/src/components/Palestrantes.vue

EXEMPLO 3: COMPONENTE EVENTOS

```
22
23  ✓    <tbody>
24  ✓      <tr v-for="evento in eventos" :key="evento.id">
25          <td>{{ evento.id }}</td>
26          <td>{{ evento.titulo }}</td>
27          <td>{{ evento.horario }}</td>
28          <td>{{ evento.local }}</td>
29      </tr>
30  </tbody>
```

/src/components/Eventos.vue

ATIVIDADE PRÁTICA 2: COMPONENTE EVENTOS

Chegou a hora de dar vida aos eventos! Utilize o componente **Eventos** para exibir uma lista dinâmica de eventos no seu `App.vue`.

INTEGRAR EVENTOS

No seu `App.vue`, importe o `Eventos.vue` e insira-o no template principal.

PREPARAR DADOS

Crie uma lista de objetos JavaScript representando os eventos, com propriedades como `titulo`, `horario`, `local` e `id`. Evento também pode ter um `titulo`



Dica: Não esqueça de configurar as `props` no componente `Eventos.vue` para receber os dados corretamente!

EXEMPLO 4: COMPONENTE MENU

```
1  <script setup lang="ts">
2  √ defineProps<{
3    |   titulo: string;
4    |   links: Array;
5    | }>();
6  </script>
7
8  √ <template>
9  √   <nav class="navbar navbar-expand-lg bg-body-tertiary border-bottom mb-4">
10 √   <div class="container">
11     <!-- Marca / título do site -->
12     <a class="navbar-brand fw-bold" href="#">{{ titulo }}</a>
```

/src/components/Menu.vue

ATIVIDADE PRÁTICA 3: COMPONENTE MENU

Implemente o componente `Menu` para criar uma barra de navegação dinâmica em seu `App.vue`, permitindo que os links sejam configurados externamente.

INTEGRAÇÃO

No seu `App.vue`, importe o `Menu.vue` e adicione-o ao template principal.

DADOS PARA O MENU

Crie uma estrutura de dados (um array de objetos) para seus itens de menu. Cada objeto deve ter as propriedades `nome` (ex: "Home"), `url` (ex: "/"), e `ativo` (boolean).

PASSE AS PROPS

Envie um título para o menu (ex: "Portal de Eventos") e o array de links criado como props para o componente `Menu`.



Dica: Certifique-se de que o `Menu.vue` esteja configurado para receber as props `título` e `links`.

ATIVIDADE PRÁTICA 4: COMPONENTE CARROSSEL

Para estruturar a navegação do portal e apresentar conteúdo visualmente, vamos criar dois novos componentes:

CARROSSEL

Um componente para exibir um carrossel de imagens responsivo, aproveitando os recursos do Bootstrap. Ele receberá dinamicamente a lista de imagens do componente pai (App.vue).

EXERCÍCIO: ANÁLISE DOS COMPONENTES EXEMPLO

Para consolidar o aprendizado sobre a estrutura e funcionalidade dos componentes Vue, responda às seguintes perguntas sobre os 6 componentes de exemplo (Botão, Palestrantes, Eventos, Menu, Curtidas e Carrosel):

→ PROPS RECEBIDAS

Quais **props** cada componente pode receber? Descreva o propósito de cada uma.

→ VARIÁVEIS REATIVAS

Quais **variáveis reativas** (`ref()`) cada componente utiliza e qual a função de cada uma?

→ DIRETIVAS VUE

Quais **diretivas** (`v-if`, `v-for`, `v-bind`, `v-on`, `v-model`) são aplicadas em cada componente e para qual finalidade?

SPA: SINGLE PAGE APPLICATION

UMA PÁGINA, INFINITAS POSSIBILIDADES

O QUE É?

Aplicação que **carrega apenas uma página HTML** e muda conteúdo dinamicamente

COMO FUNCIONA?

vue-router gerencia navegação sem recarregar o navegador

POR QUE USAR?

Navegação rápida, experiência fluida, código modular

BENEFÍCIOS DAS SPAS



PERFORMANCE

Navegação instantânea entre páginas sem recarregar tudo



EXPERIÊNCIA

Interface mais responsiva e aplicação mais fluida



ORGANIZAÇÃO

Cada página é um componente reutilizável

VUE ROUTER: CONFIGURAÇÃO

INSTALAÇÃO

```
npm install vue-router
```

NO MAIN.TS:

```
import { createApp } from 'vue'
import App from './App.vue'
import router from './router'

const app = createApp(App)
app.use(router)
app.mount('#app')
```



Exemplo (SPA completo): <https://stackblitz.com/edit/projeto-aula-spa?file=src%2Frouter%2Findex.js>

VUE ROUTER: ENTENDENDO O `INDEX.TS`

O arquivo `src/router/index.ts` é o coração da sua navegação Vue, centralizando a definição de todas as rotas da sua Single Page Application.

VIEWS

Views serão as páginas da aplicação. Serão componentes comuns que irão chamar outros componentes para formular uma página. Diretório `/src/views`

IMPORTAÇÃO DO ROUTER DO ROUTER E VIEWS

A importação dos elementos de criação e configuração do router e das views para o index

```
1  import { createRouter, createWebHistory } from 'vue-router';  
2  
3  import Home from '../views/Home.vue';  
4  import Evento from '../views/Evento.vue';
```

VUE ROUTER: ENTENDENDO O `INDEX.TS`

CONFIGURAÇÃO DAS ROTAS COM PATH E COMPONENT

path: Será como aparece na URL a página e a forma de referenciá-la

component: Componente que representa a página

```
6  ✓ const routes = [  
7    { path: '/', component: Home },  
8    { path: '/eventos', component: Evento },  
9  ];  
10  
11  ✓ const router = createRouter({  
12    history: createWebHistory(),  
13    routes,  
14  });  
15  
16  export default router;
```

ROUTERLINK E ROUTERVIEW

O `<RouterLink>` é o componente preferencial para criar links de navegação. Ele é renderizado como uma tag `<a>`, mas intercepta o clique para evitar um recarregamento completo da página, permitindo uma transição suave entre as rotas da sua SPA.

ROUTERVIEW: ONDE AS ROTAS GANHAM VIDA

O `<RouterView>` funciona como um "slot" dinâmico. É neste local que o componente correspondente à rota atual será renderizado. À medida que o usuário navega, o Vue Router atualiza o conteúdo dentro do `<RouterView>` para exibir a "view" da página correta.

Você pode ver o exemplo completo do `App.vue` aqui: <https://stackblitz.com/edit/projeto-aula-spa?file=src%2Frouter%2FNovoApp.vue>

EXERCÍCIO FINAL: NAVEGAÇÃO COM VUE ROUTER

Para solidificar seu conhecimento, crie uma navegação completa entre as páginas do portal de eventos.



1. CONFIGURAR ROTAS

Mapeie os caminhos URL para os componentes de 'Início', 'Eventos' e 'Sobre' em `src/router/index.ts`.



2. CRIAR LINKS DE NAVEGAÇÃO

No componente `MenuPrincipal.vue`, utilize `<RouterLink>` para cada item do menu, direcionando para suas rotas.



3. RENDERIZAR VISTAS

No `App.vue`, insira o `<RouterView>` para exibir o conteúdo das rotas ativas dinamicamente.

OBRIGADO!

Para mais informações:

040601985@prof.unama.br

