

# Trilha 05

*Estilização com Bootstrap e Vue.js*

## 4 - Estilização com Bootstrap e Vue.js

O **Bootstrap** é um dos frameworks CSS mais populares e amplamente utilizados no desenvolvimento web. Ele fornece uma ampla variedade de componentes e um sistema de grid responsivo que facilita a criação de interfaces modernas e adaptáveis. Quando integrado ao **Vue.js**, o Bootstrap permite desenvolver aplicações altamente responsivas e visualmente atraentes com facilidade.

---

### 4.1 Bootstrap e Vue.js

#### O que é o Bootstrap?

O **Bootstrap** é um framework CSS open-source que simplifica o design de páginas web, fornecendo uma coleção de componentes reutilizáveis e um sistema de layout baseado em **Flexbox** e **CSS Grid**.

#### Por que integrar o Bootstrap ao Vue.js?

O Vue.js é um framework reativo que facilita a criação de componentes interativos, enquanto o Bootstrap oferece um design pré-construído que melhora a aparência das aplicações. A integração dos dois permite:

- ✓ **Criação rápida de interfaces responsivas.**
- ✓ **Uso de componentes prontos e estilizados.**
- ✓ **Menos código CSS customizado.**
- ✓ **Adoção de padrões modernos de UI/UX.**

---

## 4.2 Integração do Bootstrap com Vue.js

Há várias formas de integrar o Bootstrap em um projeto Vue.js. Vamos explorar as principais abordagens.

### 1 Instalando Bootstrap via NPM

A maneira mais recomendada de usar o Bootstrap em projetos Vue.js modernos é instalá-lo via **NPM**:

```
npm install bootstrap
```

Após a instalação, importe o CSS globalmente no `main.js`:

```
import 'bootstrap/dist/css/bootstrap.min.css';
import 'bootstrap/dist/js/bootstrap.bundle.min.js';
```

## 2 Adicionando Bootstrap via CDN

Se você deseja uma integração rápida sem instalar dependências, pode incluir o Bootstrap via CDN no seu arquivo `index.html`:

```
<head>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</head>
```

## 4.3 Uso de Classes Utilitárias e Grid System

O Bootstrap fornece uma série de **classes utilitárias** que simplificam a estilização e um **Grid System** altamente flexível baseado em **Flexbox**.

---

### Classes Utilitárias

As **classes utilitárias** do Bootstrap permitem personalizar elementos rapidamente sem a necessidade de escrever CSS adicional.

#### Exemplo de Classes Utilitárias

```
<div class="p-3 bg-primary text-white text-center">
  Caixa Azul com Padding e Texto Branco
</div>
```

- `p-3`: Define um espaçamento interno (padding).
- `bg-primary`: Define um fundo azul (cor primária).
- `text-white`: Define a cor do texto como branco.
- `text-center`: Centraliza o texto.

---

### Grid System

O **Grid System** do Bootstrap divide a tela em **12 colunas**, permitindo a criação de layouts altamente responsivos.

### Exemplo de Layout Responsivo

```
<div class="container">
  <div class="row">
    <div class="col-md-4 bg-light p-3">Coluna 1</div>
    <div class="col-md-4 bg-secondary text-white p-3">Coluna 2</div>
    <div class="col-md-4 bg-dark text-white p-3">Coluna 3</div>
  </div>
</div>
```

- **col-md-4**: Cada coluna ocupa 4 das 12 colunas disponíveis na tela.
- **row**: Define uma linha flexível.
- **container**: Garante espaçamento adequado na tela.

## 4.4 Criação de Layouts Responsivos

O Bootstrap facilita a criação de **layouts adaptáveis**, que se ajustam automaticamente a diferentes tamanhos de tela.

### Exemplo: Layout Responsivo com Vue.js

```
<template>
  <div class="container">
    <div class="row">
      <div class="col-lg-6 col-md-12 p-4 border">Seção 1</div>
      <div class="col-lg-6 col-md-12 p-4 border">Seção 2</div>
    </div>
  </div>
</template>
```

- Em telas **grandes (lg)**, os elementos ocupam **6 colunas** cada.
- Em telas **médias (md)**, cada elemento ocupa **12 colunas**, ficando em **linha única**.

## 4.5 Componentes Básicos

O Bootstrap oferece uma série de **componentes reutilizáveis**. Vamos explorar alguns dos principais.

---

### Navbars

As **barras de navegação** (Navbars) são amplamente usadas em aplicações web.

## Exemplo de Navbar

```
<template>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="#">Meu Site</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="menu">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item"><a class="nav-link" href="#">Home</a></li>
          <li class="nav-item"><a class="nav-link" href="#">Sobre</a></li>
        </ul>
      </div>
    </div>
  </nav>
</template>
```

## Botões

Os botões no Bootstrap podem ser personalizados com diferentes cores e tamanhos.

### Exemplo de Botões

```
<template>
  <button class="btn btn-primary">Botão Azul</button>
  <button class="btn btn-danger btn-lg">Botão Grande Vermelho</button>
</template>
```

## Cards

Os **Cards** são usados para exibir informações organizadas.

### Exemplo de Card

```
<template>
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Título do Card</h5>
      <p class="card-text">Descrição do card.</p>
      <button class="btn btn-primary">Saiba Mais</button>
    </div>
  </div>
</template>
```

## Modais

Os **Modais** são usados para exibir conteúdos em uma janela flutuante.

### Exemplo de Modal

```
<template>
  <button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#meuModal">
    Abrir Modal
  </button>

  <div class="modal fade" id="meuModal" tabindex="-1">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title">Título do Modal</h5>
          <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
        </div>
        <div class="modal-body">Conteúdo do Modal.</div>
        <div class="modal-footer">
          <button class="btn btn-secondary" data-bs-dismiss="modal">Fechar</button>
        </div>
      </div>
    </div>
  </div>
</template>
```

## 4.6 Personalização de Estilos com SCSS

O Bootstrap permite a personalização avançada por meio do **SCSS**.

Passo 1: Instalar o SCSS

```
npm install sass
```

Passo 2: Criar um Arquivo `custom.scss`

```
$primary: #ff5733; // Cor personalizada
@import "bootstrap/scss/bootstrap";
```

Passo 3: Importar no Projeto

```
import './assets/custom.scss';
```

Agora, os componentes do Bootstrap terão a **cor primária personalizada**!

## Material Complementar: Documentação Oficial do Bootstrap Vue

Para aprofundar seus conhecimentos sobre a integração do **Bootstrap com Vue.js**, é altamente recomendável explorar a **documentação oficial do Bootstrap Vue**, disponível em:

 <https://bootstrap-vue.org/>

A documentação oficial do **Bootstrap Vue** contém informações detalhadas sobre a utilização do Bootstrap dentro do ecossistema Vue.js, incluindo:

- ✓ **Guia de Instalação:** Passo a passo para configurar o Bootstrap Vue em um projeto Vue.js.
- ✓ **Componentes e APIs:** Explicação completa sobre cada componente disponível, incluindo suas propriedades e eventos.
- ✓ **Layouts Responsivos:** Uso avançado do Grid System para criação de layouts dinâmicos e adaptáveis.
- ✓ **Formulários e Validação:** Implementação de formulários interativos e validação com Bootstrap Vue.
- ✓ **Personalização de Estilos:** Configuração de temas personalizados utilizando SCSS e variáveis do Bootstrap.
- ✓ **Exemplos Práticos:** Códigos prontos para ajudar no desenvolvimento de projetos reais.

## Por que consultar a documentação do Bootstrap Vue?

- 📌 **Conteúdo atualizado:** Mantido por desenvolvedores oficiais e pela comunidade.
- 📌 **Referência completa:** Ideal tanto para iniciantes quanto para desenvolvedores avançados.
- 📌 **Exemplos interativos:** Permite testar diferentes implementações diretamente no navegador.

## Conclusão

A documentação oficial do **Bootstrap Vue** é um recurso essencial para quem deseja dominar a estilização e criação de interfaces responsivas utilizando **Bootstrap no Vue.js**. Consultá-la frequentemente ajudará você a aprimorar suas habilidades e desenvolver aplicações mais robustas e eficientes! 🚀

## Lista de Atividades - Estilização com Bootstrap e Vue.js

---

### Questões Teóricas (10 questões)

1. O que é o Bootstrap e quais são suas principais vantagens ao ser utilizado no desenvolvimento web com Vue.js?
  2. Quais são as principais formas de integrar o Bootstrap a um projeto Vue.js? Explique as diferenças entre utilizar CDN e instalar via NPM.
  3. Explique o conceito de classes utilitárias no Bootstrap. Como elas ajudam a estilizar elementos sem a necessidade de CSS personalizado?
  4. Descreva o funcionamento do Grid System do Bootstrap. Como ele facilita a criação de layouts responsivos?
  5. Qual é a importância das classes `col-`, `row`, `container` e `container-fluid` no sistema de grid do Bootstrap?
  6. Explique a diferença entre os componentes do Bootstrap `Navbar`, `Card` e `Modal`. Quando utilizar cada um deles?
  7. O que são `breakpoints` no Bootstrap e como eles afetam o comportamento dos layouts responsivos?
  8. O Bootstrap permite personalizar seus estilos utilizando SCSS. Qual a vantagem de utilizar SCSS ao invés de CSS puro para modificar estilos no Bootstrap?
  9. Qual a importância dos modais em uma aplicação Vue.js? Cite um exemplo prático de uso em um sistema web.
  10. Como o Vue.js pode ser combinado com o Bootstrap para criar interfaces mais dinâmicas e interativas?
- 

### Questões Práticas (10 questões)

1. Crie um layout básico utilizando o Grid System do Bootstrap, com três colunas de tamanhos diferentes para telas grandes e pequenas.
2. Crie uma Navbar responsiva utilizando o Bootstrap, contendo um menu de navegação com três links e um botão de alternância para dispositivos móveis.
3. Implemente um botão estilizado (`v-btn`) que altere sua cor dinamicamente ao ser clicado, utilizando classes do Bootstrap.
4. Crie um Card (`v-card`) contendo uma imagem, um título e um botão de ação. O botão deve exibir um alerta ao ser pressionado.
5. Desenvolva um layout de página utilizando `container`, `row` e `col-`, criando uma estrutura que se adapte automaticamente para telas menores.
6. Implemente um Modal Bootstrap que seja ativado ao clicar em um botão. O modal deve conter um título, uma mensagem e um botão para fechá-lo.
7. Crie um formulário responsivo utilizando Bootstrap com campos para Nome, E-mail e Senha, aplicando validação de entrada nos campos.



8. Personalize o Bootstrap utilizando SCSS para alterar a cor primária dos botões e modificar a aparência de um Card.
9. Crie uma página Vue.js estilizada com Bootstrap, contendo um cabeçalho (Navbar), um conteúdo centralizado e um rodapé fixo.
10. Utilizando Vue.js, crie um sistema de abas (Tabs) onde ao clicar em cada aba, o conteúdo seja alterado dinamicamente utilizando classes do Bootstrap.

- *Responda todas as questões teóricas e práticas.*
- *Organize as respostas em um arquivo PDF, contendo nome e data.*
- *Envie o PDF aos professores responsáveis, seguindo o padrão de nomenclatura.*
- *Valide o material com um professor antes de prosseguir.*
- *Certifique-se de cumprir o prazo de entrega.*

**Padrão de nomenclatura**

**NomeCompleto\_TrilhaX\_DataDeEntrega.pdf**

**Exemplo: JoãoSilva\_Trilha1\_2025-01-30.pdf**